Dell HPC NFS Storage Solution High Availability Configurations with Large Capacities

High availability NSS configurations for capacities greater than 100TB.

Xin Chen, Garima Kochhar and Mario Gallegos

Dell HPC Engineering

Version 2.1, May 2012

This document is for informational purposes only and may contain typographical errors and technical inaccuracies. The content is provided as is, without express or implied warranties of any kind.

© 2012 Dell Inc. All rights reserved. Dell and its affiliates cannot be responsible for errors or omissions in typography or photography.

Dell, the Dell logo, and PowerEdge are trademarks of Dell Inc. Intel and Xeon are registered trademarks of Intel Corporation in the U.S. and other countries. Microsoft, Windows, and Windows Server are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

May 2012| Rev 2.1

Contents

Executive summary (Updated May 2012) 6
1. Introduction 7
2. NSS-HA solution review 7
2.1. Availability in NSS-HA 9
3. NSS-HA architecture 10
3.1. Storage in NSS-HA 11
3.2. Potential failures and fault tolerant mechanisms in NSS-HA 13
4. New components and updates from previous versions of the solution 14
4.1. Storage density 14
4.2. Storage configuration 15
4.3. Red Hat Enterprise Linux operating system 15
4.4. Red Hat scalable file system package 16
4.5. Summary of changes 16
5. Evaluation 17
5.1. Method 17
5.2. Test bed 18
5.3. Tuning options 22
5.4. Functionality tests 23
6. Performance tests 25
6.1. InfiniBand sequential writes and reads 26
6.2. 10 Gigabit Ethernet sequential writes and reads 27
6.3. Random writes and reads 29
6.4. Metadata tests 31
6.5. NFSv3 compared to NFSv4 34
7. Conclusion 36
8. References 37
Appendix A: NSS-HA Recipe (Updated May 2012) 38
A.1. Pre-install preparation 39
A.1.1. NSS-HA cluster specification 40
A.1.2. Checklist 42
A.2. Server hardware setup 43
A.2.1. Checklist 44
A.3. Server software configuration 45
A.3.1. Install RHEL 6.1, configure swap disks, and install XFS packages. 45

A.3.2. Configure Multipath 46
A.3.3. Install Mellanox OFED package and set network IPs 48
A.3.4. Install Operating system and storage management tools 49
A.3.5. Network security setting 50
A.3.6. Configure startup services. 51
A.3.7. Configure fence devices 51
A.3.8. Checklist 53
A.4. Performance tuning on the server 54
A.4.1. Checklist 55
A.5. Storage hardware setup 56
A.6. Storage configuration 59
A.6.1. Checklist 61
A.7. NSS HA cluster setup 62
A.7.1. Prepare 62
A.7.2. Create HA cluster 63
A.7.3. Verification 65
A.8. Quick test of HA setup 68
A.9. Useful commands and references 69
A.9.1. Manually modify cluster configuration file 69
A.9.2. Manually stop, disable, start and relocate the cluster service 69
A.9.3. Debug HA cluster configuration 70
A.9.4. Remove a node from HA cluster 70
A.9.5. Configure the shared storage array manually 71
A.10. Performance tuning on clients 74
A.11. Scripts 75
Appendix B: Benchmarks and test tools 76
B.1. IOzone 76
B.2. mdtest 78
B.3. Checkstream 79
B.4. dd 80

Figures

Figure 1. Overview of the NSS-HA solution 8
Figure 2. A failure scenario in NSS-HA 9
Figure 3. NSS-HA architectural diagram 10
Figure 4. NFS server configuration in NSS-HA 11
Figure 5. Steps to configure storage 12
Figure 6. Storage and file system layout 15
Figure 7. NSS-HA test bed 19
Figure 8. InfiniBand large sequential write performance 26
Figure 9. InfiniBand large sequential read performance 27
Figure 10. 10GbE large sequential write performance 28
Figure 11. 10GbE large sequential read performance 28
Figure 12. Single 10gbE client sequential performance 29
Figure 13. InfiniBand random write performance 30
Figure 14. InfiniBand random read performance 31
Figure 15. InfiniBand file create performance 32
Figure 16. InfiniBand file stat performance 33
Figure 17. InfiniBand file remove performance 33
Figure 18. InfiniBand NFSv3 and NFSv4 file create performance 34
Figure 19. InfiniBand NFSv3 and NFSv4 sequential performance 35
Figure 20. InfiniBand NFSv3 and NFSv4 random performance 35

Tables

Table 1. NSS-HA mechanisms to handle failures 13
Table 2. Sample storage capacities 14
Table 3. New storage components in this release 16
Table 4. New server components in this release 17
Table 5. NSS-HA hardware configuration details 20
Table 6. NSS-HA software configuration details 21
Table 7. NSS-HA firmware and driver configuration details 21
Table 8. NSS-HA client configuration details 22

Executive summary (Updated May 2012)

This solution guide describes the large capacity configurations of the Dell HPC NFS Storage Solution with high availability support (NSS-HA). It presents an architecture overview, and provides tuning best practices and performance details for configurations with capacities of 144TB and 288TB. These configurations break the 100TB limit of previous supported configurations.

The NSS-HA solutions described here are designed for high availability using a pair of NFS servers in active-passive configuration to provide storage service to the HPC compute cluster. As in previous versions of this solution guide, the goal is to improve service availability and maintain data integrity in the presence of possible failures or faults, and to maximize performance in the failure-free case.

May 2012 update

All changes are in Appendix A: NSS-HA Recipe (Updated May 2012) and the attached scripts. Version 2.1 of this document contains updated instructions for the multipath set-up, more details on OMSA and MDSM configuration and a modification to some of the scripts. The recipe contains a checklist for each section as well

1. Introduction

This Solution Guide provides information on the latest Dell NFS Storage Solution high availability configurations (NSS-HA). The NSS-HA uses the NFS file system along with the Red Hat Scalable File system (XFS) and Dell PowerVault storage to provide an easy to manage, reliable and cost effective storage solution for HPC clusters. With this latest offering, NSS-HA configurations are now available in configurations greater than a 100 Terabytes.

The philosophy and design principles for this release remain the same as previous Dell NSS-HA configurations. Hence this version of the solution guide primarily describes the deltas in configuration and performance. For complete details, review this document along with the previous version titled "Dell HPC NFS Storage Solution High Availability Configurations, Version 1.1.". The main changes in this version of the solution include support for larger capacities. Also presented is the associated performance characterization and updated software versions.

The following sections describe the technical details, evaluation method and the expected performance of the solution. An extensive appendix provides a complete set of instructions on the configuration steps and tuning parameters required to deploy such a solution.

2. NSS-HA solution review

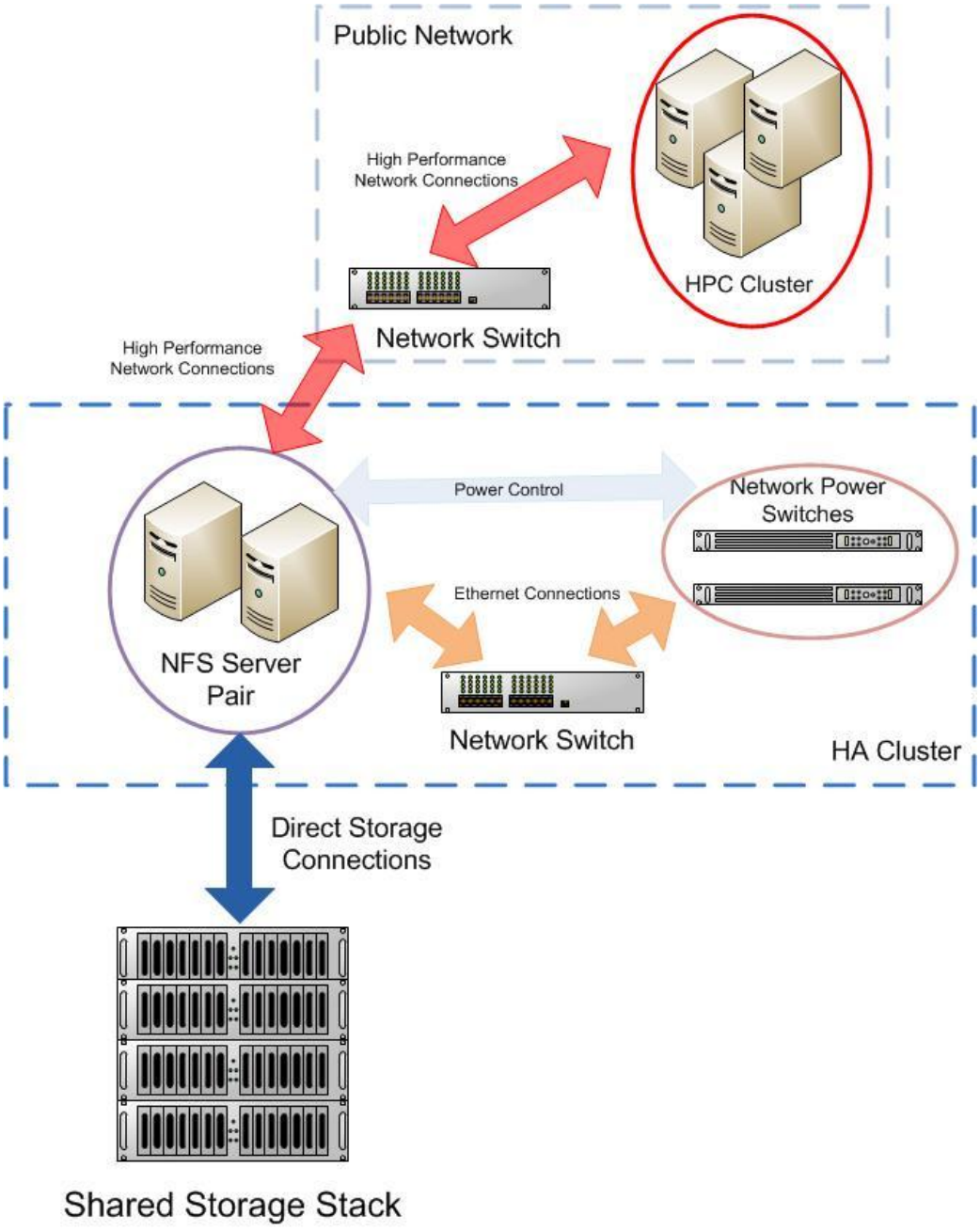
The design of this version of the NSS-HA solution is similar to previous versions. This section provides a quick review of the NSS-HA solution. Complete details are available in the document "Dell HPC NFS Storage Solution High Availability Configurations, Version 1.1". This section can be skipped for readers who are already familiar with the NSS-HA architecture.

Figure 1 depicts the general overview of the NSS-HA solution. The core of the solution is a high availability (HA) cluster, which provides a highly reliable and available storage service to the HPC compute cluster via a high performance network connection such as InfiniBand (IB) or 10 Gigabit Ethernet (10GbE). The HA cluster has shared access to disk-based Dell PowerVault storage in a variety of capacities.

The HA cluster consists of several components as listed below:

- High Availability nodes - These are servers configured with the Red Hat Enterprise Linux high availability cluster software stack. In the NSS-HA solution, two systems are deployed as a pair of NFS severs; they are configured in an active/passive mode, and have direct access to the shared storage stack.
- Network switch for the HA cluster (or the private network) – The private network is used for communication between the HA cluster nodes and other cluster hardware such as network power switches and the fence devices which are installed in the cluster nodes.
- Fence devices – Fence devices are required for fencing (rebooting) the failed or misbehaving cluster node in the HA cluster. In the NSS-HA solution, two types of fence devices are configured: Switched Power Distribution Units (PDU) and the Dell server management controller, the iDRAC.

Figure 1. Overview of the NSS-HA solution



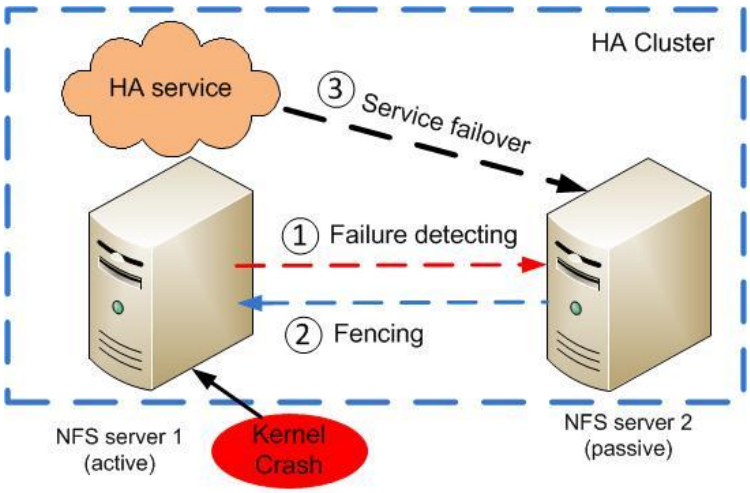
2.1. Availability in NSS-HA

A major goal of the NSS-HA solution is to improve storage service availability in the presence of possible failures or faults. This goal is achieved by a **"failover"** process implemented by Red Hat Enterprise High Availability Cluster software stack.

Figure 2 shows a typical scenario of how storage service availability is guaranteed in the NSS-HA solution. In this scenario, assume a kernel crash occurs on an NFS server (the active one) which is the NFS gateway for the compute cluster. The service availability is protected by three steps:

1) Failure detection – Resources related to the storage service, such as file system, service IP address, etc., are defined, configured and monitored for health by the HA cluster. Any interruption in access to the storage will be detected. In this case, once a kernel crash occurs at NFS server 1 (the active one), a message in terms of loss of heartbeat signal will pass to NFS server 2, and server 2 will recognize that the server 1 has failed.
2) Fencing – In the HA cluster, once a node notices that the other node has failed, it will fence (reboot) the failed node via a fence device. This is to ensure that only one server accesses the data at any point to protect data integrity. In NSS-HA, a node can fence the other via the Dell iDRAC or an APC PDU. The fence devices and corresponding fence commands are configured as part of the HA cluster configuration process. In this case, NFS server 2 will fence NFS server 1.
3) Service failover – In the HA cluster, only after a node successfully fences the other can the service failover process be started. Failover means that the HA service running previously on the failed server will be now transferred to the healthy one. In this case, once NFS server 2 has successfully fenced server 1, the HA service will be transferred to and started on NFS server 2.

Figure 2. A failure scenario in NSS-HA



From the perspective of the compute cluster, there will be degradation in performance during the actual HA failover process. But the failover is transparent to the compute cluster as far as possible and user applications continue to function and access data as before.

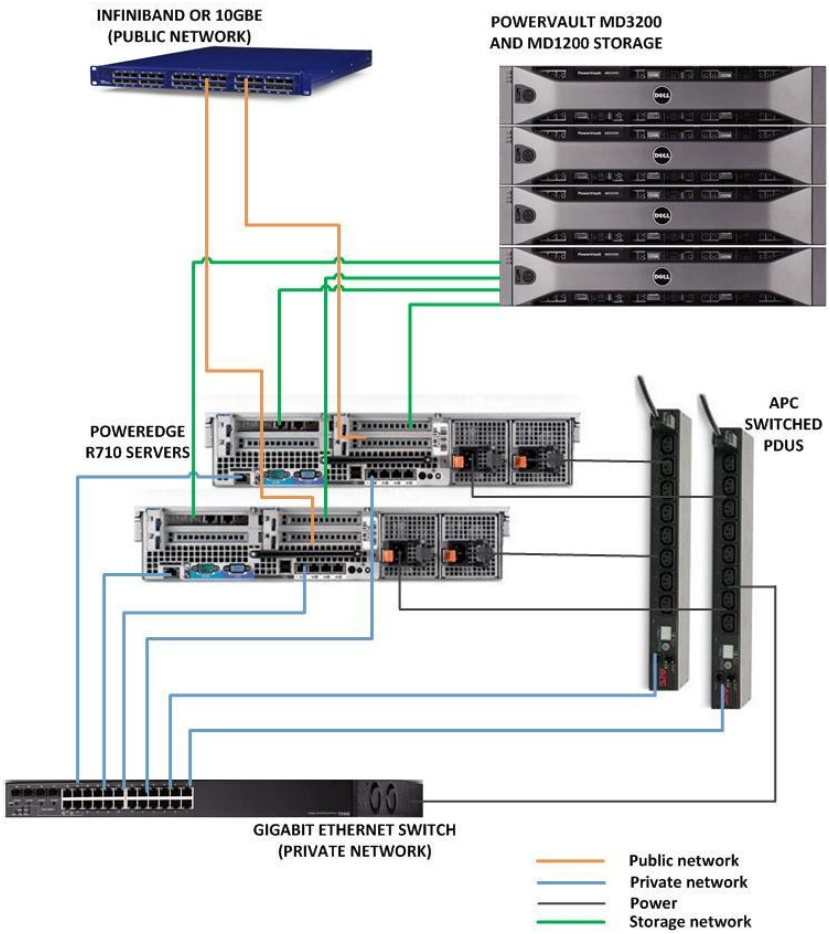
The HA service can be defined and configured in the cluster configuration process. In the NSS-HA, NFS export, the service IP via which the compute nodes access the NFS server, and LVM are configured as a HA service.

Appendix A: NSS-HA Recipe includes detailed instructions on the configuration steps.

3. NSS-HA architecture

Figure 3 presents the architectural diagram of the NSS-HA solution. A pair of PowerEdge R710 servers are configured as an active-passive HA pair and function as an NFS gateway for the HPC compute cluster (also called the clients).

Figure 3. NSS-HA architectural diagram

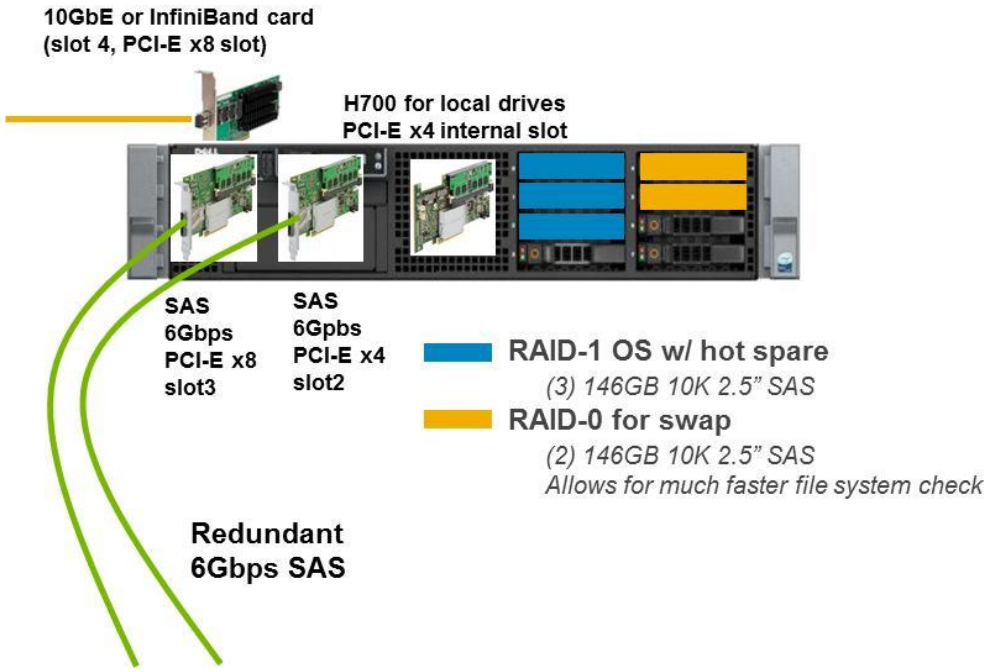


Both NFS servers are connected to a shared PowerVault disk storage at the backend. The user data resides on an XFS file system created on this storage. The XFS file system is exported via NFS to the clients.

The NFS servers are connected to the clients via the public network. This network can be either InfiniBand or 10 Gigabit Ethernet. Configuration of the NFS server is shown in Figure 4. It is also discussed in Section 5.2 that describes the Test bed used to assess this solution.

For the HA functionality of the NFS servers, a private Gigabit Ethernet network is configured to monitor server health and heartbeat, and to provide a route for the fencing operations. Power to the NFS servers is driven by two APC switched PDUs on two separate power buses.

Figure 4. NFS server configuration in NSS-HA



The NSS-HA architecture is discussed in detail in the previous version of this solution guide [4].

3.1. Storage in NSS-HA

The NSS-HA is a storage solution, in which a shared storage array is directly connected to the HA cluster nodes, as shown in Figure 1 and Figure 3. Access to the storage is provided to users via the HA service defined in the HA cluster. This section provides general information about the NSS-HA storage configuration. Appendix A: NSS-HA Recipe includes detailed instructions on the storage and file system configuration for NSS-HA. Figure 5 lists the steps for deploying a storage stack in NSS-HA. There are four steps in total:

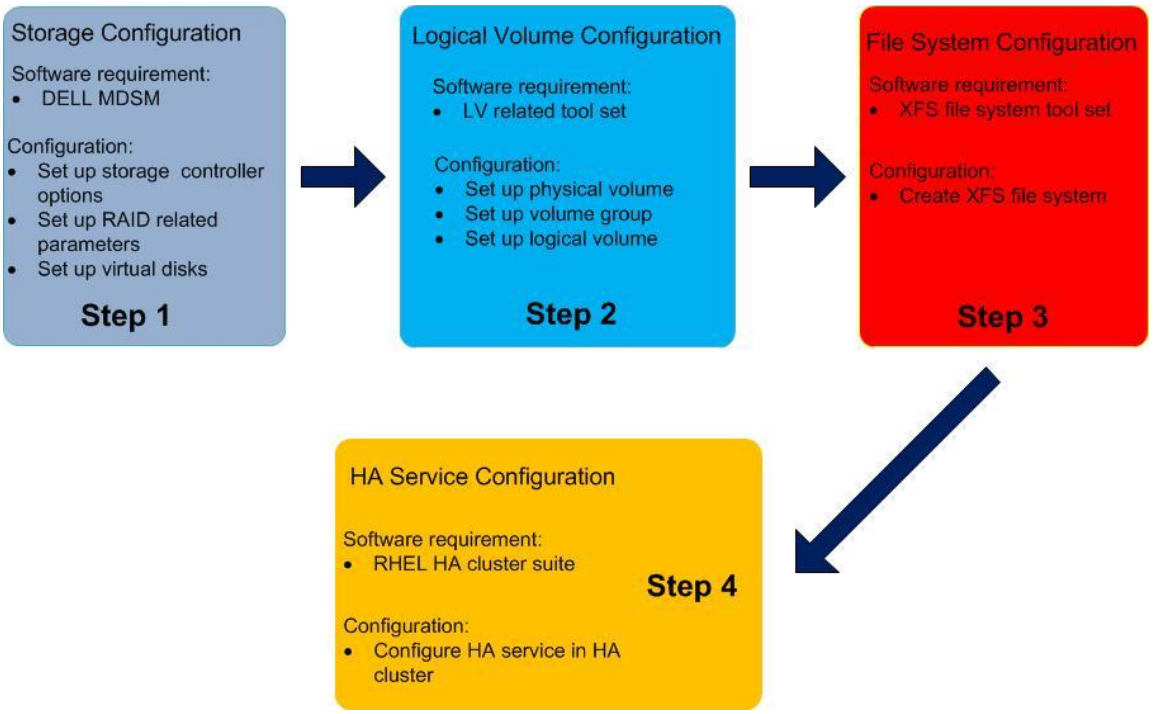
1) Step 1 – Storage configuration

   In this step, the storage array is configured. As NSS-HA uses the Dell PowerVault storage arrays, Dell Modular Disk Storage Manager (MDSM) is required for the configuration. For the purpose of ensuring data integrity as much as possible, RAID 6 is adopted and RAID controller caching mirroring is enabled on the storage controllers. The RAID 6 based virtual disks are created on each of the storage arrays used in the solution. For example, if four storage arrays are included in the solution, there are four virtual disks created.

2) Step 2 – Logical volume configuration

In this step, a logical volume is created to access the capacity configured on the storage arrays. NSS-HA requires a simple way to manage and scale a storage stack and so Linux logical volume manager is used for its simplicity. In order to create a logical volume, physical volumes (PVs) are created first. The PVs have a one-to-one correspondence with the virtual disks created in the previous step. Once the physical volumes are created, they are combined into a volume group (VG), and then a logical volume is allocated from the disk space of the VG. The LV related command sets are part of the base Red Hat Enterprise Linux Operating System.

Figure 5. Steps to configure storage



3) Step 3 – File system configuration

In this step, a file system is created on the logical volume created in step 2. In NSS-HA, the Red Hat Scalable file system (XFS file system) is adopted, as its features such as quick recovery, massive scalability, and high performance [(1)] satisfy the high reliability, availability, and performance goals of NSS-HA. The Scalable file system package is distributed with Red Hat Enterprise Linux Operating System as an add-on component.

4) Step 4 – HA service configuration

In this step, the logical volume and XFS file system created in the previous two steps are defined as resources in HA cluster configuration file. The XFS file system is exported via NFS, and the NFS export is also defined as a resource in HA cluster. An HA service is built using these resources and this HA service is what provides access to the storage. The HA cluster suite is an add-on component of the Red Hat Enterprise Linux Operating System.

3.2. Potential failures and fault tolerant mechanisms in NSS-HA

In the real world, there are many different types of failures and faults which can impact the functionality of NSS-HA. Table 1 lists the potential failures which can be tolerated in an NSS-HA solution based on the architecture described in Section 3. The analysis below assumes that the HA cluster service is running on the "active" server, the "passive" server is the other component of the cluster.

Table 1. NSS-HA mechanisms to handle failures

Failure type	Mechanism to handle failure
Single local disk failure on a server	Operating system installed on a two-disk RAID 1 device with one hot spare. Single disk failure is unlikely to bring down server.
Single server failure	Monitored by the cluster service. Service fails over to passive server.
Power supply or power bus failure	Dual power supplies in each server. Each power supply connected to a separate power bus. Server will continue functioning with a single power supply.
Fence device failure	iDRAC used as primary fence device. Switched PDUs used as secondary fence devices.
SAS cable/port failure	Two SAS cards in each NFS server. Each card has a SAS cable to storage. A single SAS card/cable failure will not impact data availability.
Dual SAS cable/card failure	Monitored by the cluster service. If all data paths to the storage are lost, service fails over to the passive server.
InfiniBand /10GbE link failure	Monitored by the cluster service. Service fails over to passive server.
Private switch failure	Cluster service continues on the active server. If there is an additional component failure, service is stopped and system administrator intervention required.
Heartbeat network interface failure	Monitored by the cluster service. Service fails over to passive server.
RAID controller failure on MD3200 storage array	Dual controllers in MD3200. The second controller handles all data requests. Performance may be degraded but functionality is not impacted.

4. New components and updates from previous versions of the solution

This section provides information on the updates in this version of the NSS-HA solution when compared to the previous version [4]. The current version includes several major changes and updates to various components of the solution.

4.1. Storage density

In previous versions of NSS-HA, each storage enclosure was equipped with 12 3.5" 2TB NL-SAS disk drives. The larger capacity 3TB disk drives are a new component in the current version. The storage arrays in the solution, Dell PowerVault MD3200 and PowerVault MD1200 expansion arrays are the same as in the previous version of the solution but with updated firmware. The higher capacity 3TB disks now allow higher storage densities in the same rack space. Table 2 provides information on new capacity configurations possible with the 3TB drives. This table is not a complete list of options; intermediate capacities are available as well.

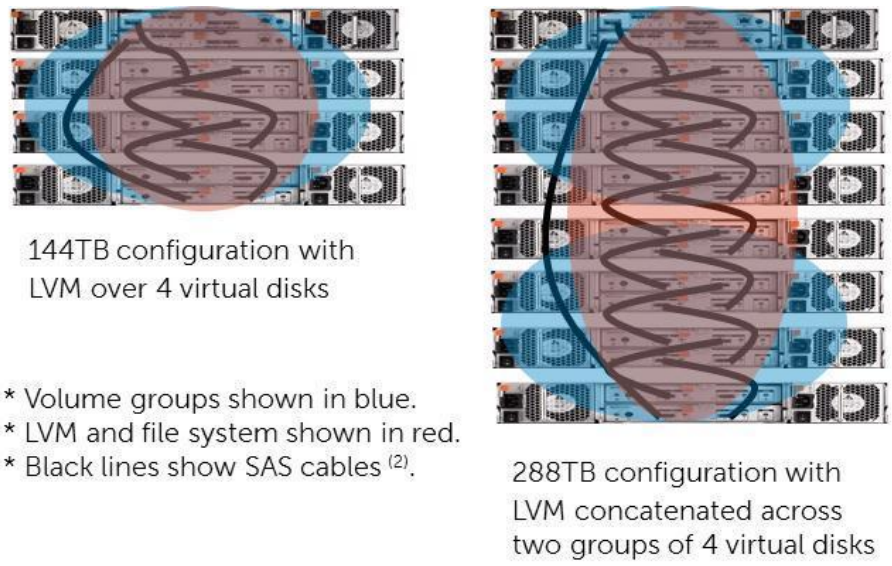
Table 2. Sample storage capacities

Sample new capacities	Storage arrays	Disk drives	Usable capacity
144TB raw capacity	4 arrays (1 MD3200 + 3 MD1200s)	48 drives (12 disks/array * 4 arrays * 3TB/disk)	109 TB
288TB raw capacity	8 arrays (1 MD3200 + 7 MD1200s)	96 drives (12 disks/array * 8 arrays * 3TB/disk)	218 TB *
* Larger capacities are possible on a custom basis. Please contact your Dell sales team for more information.			

4.2. Storage configuration

In previous versions of the solution, the file system had a maximum of four virtual disks. A Linux physical volume was created on each virtual disk. The physical volumes were grouped together into a Linux volume group and a Linux logical volume was created on the volume group. The XFS file system was created on this logical volume.

Figure 6. Storage and file system layout


144TB configuration with
LVM over 4 virtual disks
* Volume groups shown in blue.
* LVM and file system shown in red.
* Black lines show SAS cables [2].
288TB configuration with
LVM concatenated across
two groups of 4 virtual disks

With this release, if the configuration includes more than four virtual disks, the Linux logical volume (LV) is extended, in groups of four, to include the additional PVs. In other words, groups of four virtual disks are concatenated together to create the file system. Data is striped across each set of four virtual disks. However it is possible to create users and directories such that different data streams go to different parts of the array and thus ensure that the entire storage array is utilized at the same time. The configuration is shown in Figure 6 for a 144TB configuration and a 288TB configuration. Additionally, with this release the SAS cabling uses the asymmetric ring cabling scheme as shown in black lines in the figure. The previous cascade cabling scheme can also be used instead.

4.3. Red Hat Enterprise Linux operating system

In previous versions of NSS-HA, RHEL 5.5 is deployed. In the current version, RHEL 6.1 is used. Compared to RHEL 5.5, a significant change within RHEL 6.1 is the HA cluster suite. The updated HA cluster suite includes the following major changes [3]:

- The cluster configuration GUI, Conga, is updated and has a new interface
- A restart-disable failure for a service is added for configuring HA service
- An independent sub tree as non-critical can be configured

Instructions to configure the HA cluster with RHEL 6.1 are listed in Appendix A: NSS-HA Recipe in this document. Since there are several significant changes in how the HA functionality is managed with this

release, do not refer to the instructions listed in the previous RHEL 5.5 NSS-HA solution guide [4] when configuring HA on RHEL 6.1 based clusters.

4.4. Red Hat scalable file system package

In previous versions of NSS-HA, the version of XFS is 2.10.2-7 which is distributed with RHEL 5.5. In the current version of NSS-HA, the version of XFS used is 3.1.1-4 and is distributed with RHEL 6.1. Dell worked with Red Hat to enable support for NSS configurations l than 100 TB using this version of XFS. Thus the most important feature of the current version of XFS for users is that it is able to support greater than a 100 Terabytes of storage capacity.

4.5. Summary of changes

This section lists the similarities and differences between this release of the NSS-HA and prior versions. Table 3 lists the similarities and difference in storage components. Table 4 lists the similarities and differences in the NFS servers. Details of the complete test bed are provided in Section 5.2.

Table 3. New storage components in this release

Storage components	Previous release[4]	This version	Reason for update
Storage array	PowerVault MD3200 and MD1200s		No update
PowerVault MD3200 firmware	-	Latest version to support 3TB drives	New MD3200 firmware needed for 3TB drive support.
Disks in storage array	2TB NL SAS	3TB NL SAS	Increased capacity, denser solution.
RAID configuration	RAID 6 10+2, segment size 512K		No update
Capacities supported	Up to 96TB in a single file system. 192TB in two file systems	Up to 288TB in a single file system. Larger capacities are possible on a custom basis. Please contact your Dell sales team for more information.	Increased capacities, denser solution.
LVM configuration for HA	HA-LVM	Clustered LVM	Clustered LVM is supported in RHEL 6.1 and is the recommended method.

Table 4. New server components in this release

Server components	Previous release[4]	This version	Reason for update
NFS server	PowerEdge R710		NA
Memory per NFS server	48 GB	96 GB	More memory to improve performance where a large cache is useful. Also to manage possible XFS repair operations on the larger capacity file system.
Operating System	RHEL 5.5 x86_64	RHEL 6.1 x86_64	Dell has worked with Red Hat to support configurations that are >100TB using XFS with RHEL 6.1.
Kernel version	2.6.18-194.el5	2.6.32-131.0.15.el6	Update to RHEL 6.1
File system (XFS) version	2.10.2-7	3.1.1-4	For XFS > 100TB support
10 Gigabit Ethernet Driver	ixgbe 2.0.44-k2	ixgbe 3.0.12-k2	Driver native to RHEL
InfiniBand driver	Mellanox OFED 1.5.1	Mellanox OFED 1.5.3-3.0.0	Latest version of Mellanox OFED at time of solution release.
SAS cards per server	1 SAS card for capacities up to 96TB. Two SAS cards for capacities of 192TB.	2 SAS cards.	2 SAS cards improve server to storage sequential read performance by up to 12%.

5. Evaluation

The architecture proposed in this white paper was evaluated in the Dell HPC lab. This section describes the test methodology and the test bed used for verification. It also contains details on the functionality tests.

The performance characterization of the solution is one of the major differences from the previous version. Hence the performance analysis is presented as a separate section later in this document.

5.1. Method

The NFS Storage Solution described in this solution guide was tested for HA functionality and performance.

A 144TB NSS-HA configuration was used to test the HA functionality of the solution. Different types of failures were introduced and the fault tolerance and robustness of the solution was measured. Section

5.4 describes these functionality tests and their results. Functionality testing was similar to work done in the previous versions of the solution [4].

A 64 node HPC cluster was used to provide I/O workload to test the performance of the NSS-HA. The performances of the 144TB and 288TB solutions were measured against this test bed for both InfiniBand and Ethernet based clients. Details of the test bed are provided in Section 5.2. Results of the performance study are presented in Section 6.

5.2. Test bed

The test bed used to evaluate the NSS-HA functionality and performance is shown in Figure 7. A 64 node HPC compute cluster was used to provide I/O traffic for the NSS.

For the NSS-HA, two PowerEdge R710 servers were used as the NFS servers. Both servers were connected to shared PowerVault MD3200 SAS storage extended with PowerVault MD1200 arrays (the diagram above shows a 144TB solution with four MD storage arrays). A PowerConnect 5424 Gigabit Ethernet switch was used as the private HA cluster network between the servers.

The NFS servers were connected to the compute cluster via InfiniBand or 10 Gigabit Ethernet. Complete configuration details are provided in Table 5, Table 6, Table 7 and Table 8.

Figure 7. NSS-HA test bed

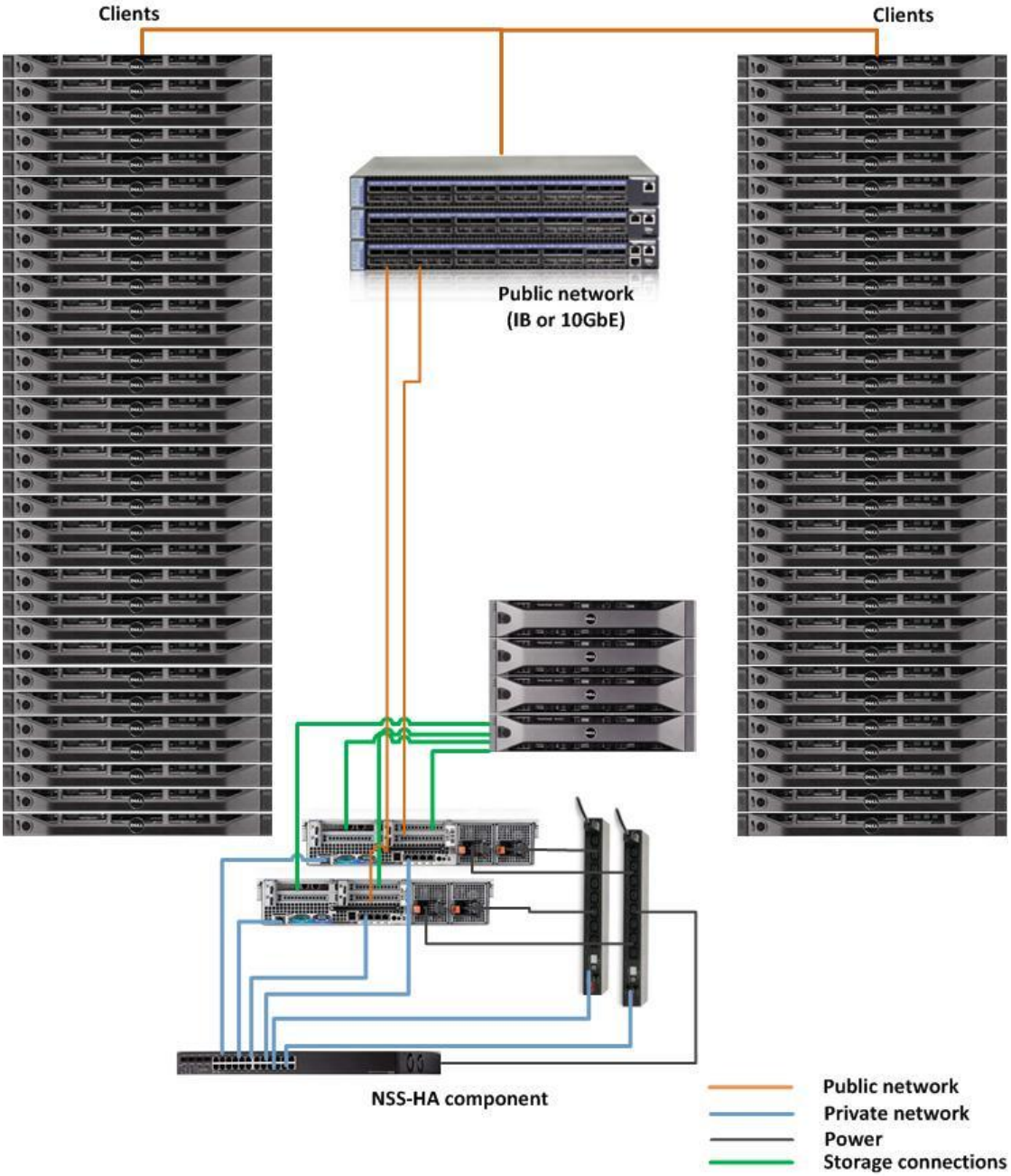


Table 5. NSS-HA hardware configuration details

Server configuration	
NFS server model	Two PowerEdge R710
Processor	Dual Intel Xeon E5630 @ 2.53GHz
Memory	12 * 4GB 1333MHz RDIMMs (The test bed used 48GB; the recommendation for production clusters is to use 96GB).
Local disks and RAID controller	PERC H700 with five 146GB 10K SAS hard drives
Optional InfiniBand HCA (slot 4)	Mellanox ConnectX-2 QDR PCI-E card
Optional 10 Gigabit Ethernet card (slot 4)	Intel X520 DA 10Gb Dual Port SFP+ Advanced
External storage controller (slot 3 and slot 2)	Two 6Gbps SAS HBA
Systems Management	iDRAC Enterprise
Power Supply	Dual PSUs
Storage configuration	
Storage Enclosure	One PowerVault MD3200 array. Three MD1200 expansion arrays for the 144TB solution. Seven MD1200 expansion arrays for the 288TB solution. High Performance Tier feature enabled on the PowerVault MD3200
RAID controllers	Duplex RAID controllers in the MD3200
Hard Disk Drives	Twelve 3TB 7200 rpm NL SAS drivers per array
Other components	
Private Gigabit Ethernet switch	PowerConnect 5424
Power Distribution Unit	Two APC switched Rack PDUs, model AP7921

Table 6. NSS-HA software configuration details

SOFTWARE	
Operating system	Red Hat Enterprise Linux (RHEL) 6.1 x86_64
Kernel version	2.6.32-131.0.15.el6 x86_64
Cluster Suite	Red Hat Cluster Suite from RHEL 6.1
File system	Red Hat Scalable File System (XFS) 3.1.1-4
Systems Management	Dell OpenManage Server Administrator 6.5.0
Storage Management	Dell Modular Disk Storage Manager 3.0.0.18

Table 7. NSS-HA firmware and driver configuration details

Firmware and Drivers	
PowerEdge R710 BIOS	6.0.7
PowerEdge R710 iDRAC	1.80.17
InfiniBand firmware	2.9.1000
InfiniBand driver	Mellanox OFED 1.5.3-3.0.0
10 Gigabit Ethernet driver	ixgbe 3.0.12-k2
PERC H700 firmware	12.10.0-0025
PERC H700 driver	megaraid_sas 00.00.05.34-rc1
6Gbps SAS firmware	07.01.08-00
6Gbps SAS driver	mpt2sas 08.101.00.00

Table 8. NSS-HA client configuration details

Client / HPC Compute Cluster	
Clients	64 PowerEdge R410 compute nodes Red Hat Enterprise Linux 6.1 x86-64
InfiniBand	Mellanox ConnectX-2 QDR HCA Mellanox OFED 1.5.3-3.0.0
InfiniBand fabric	All clients connected to a single large port count InfiniBand switch (Mellanox IS5100). Both R710 NSS-HA servers also connected to the InfiniBand switch.
Ethernet	Onboard 1 GbE Broadcom 5716 network adapter. bnx2 driver v2.1.6.
Ethernet fabric	Two sets of 32 compute nodes connected to two PowerConnect 6248 Gigabit Ethernet switches. Both PowerConnect 6248 switches have four 10GbE links each to a 10GbE PowerConnect 8024 switch. Both R710 NSS-HA servers connected directly to the PowerConnect 8024 switch. Flow control was disabled on the PowerConnect 8024 switch and two PowerConnect 6248 switches.

5.3. Tuning options

The tuning options and design choices in this version of the NSS-HA solution are similar to those in the previous version [4] of the solution. The design of this solution emphasizes data reliability and availability sometimes at the expense of performance. For custom configuration, some of these options may not apply. Analysis of NFSv3 and NFSv4 is new to this release and is discussed in detail. For other options a quick summary is provided here, detailed explanations can be found in the Solution Guide titled "Dell HPC NFS Storage Solution High Availability Configurations, Version 1.1".

Appendix A: NSS-HA Recipe provides instructions on how these tuning options should be configured.

Storage array configuration

1) Each storage array is configured with twelve 3.5" 3TB NearLine SAS disks.
2) Virtual disks are created using RAID 6, with 10 data disks and 2 parity disks.
3) Virtual disks are created with a segment size of 512k. This value should be set based on the expected application I/O profile for the cluster.
4) Cache block size on the RAID controller is set to 32k to maximize performance. This value should be set based on the expected application I/O profile for the cluster.
5) The read and write caches on the RAID controller are enabled.
6) Write cache mirroring is enabled between the two PowerVault MD3200 RAID controllers to protect data if there is a controller failure. Cache mirroring between the controllers ensures that the second controller can complete the writes to disk.

NFS Server configuration

7) The XFS file system is mounted with the wsync option.
8) The XFS file system is exported using the NFS sync option.
9) Number of concurrent NFS threads is increased from a default of 8 to 256 on the NFS servers.
10) The default OS scheduler is changed from cfq to deadline.
11) MTU is set to 9000 on the 10 Gigabit Ethernet networks.
12) The default NFS protocol used to export the XFS file system is configured to be version 3. Performance analysis of NFSv3 compared to NFSv4 showed NFSv3 to be significantly better performing for some cases. If the security enhancements of NFSv4 are preferable, it can be used at the cost of better performance. Section 6.5 discusses the performance difference between v3 and v4. Appendix A: NSS-HA Recipe includes details on how to set the NFS protocol version for the NSS-HA solution.

5.4. Functionality tests

The HA functionality of the solution was tested by simulating several component failures. The design of the tests and the test results are similar to previous versions of the solution since the broad architecture of the solution has not changed with this release. A quick summary is provided in this section. For detailed explanations refer to the Solution Guide titled "Dell HPC NFS Storage Solution High Availability Configurations, Version 1.1".

Functionality was verified for an NFSv3 as well as NFSv4 based solution.

The following failures were simulated on the cluster.

1) Server failure
2) Heartbeat link failure
3) Public link failure
4) Private switch failure
5) Fence device failure
6) One SAS link failure
7) Multiple SAS link failures

This section briefly outlines the NSS-HA response to these failures. Details on how to configure the solution to handle these failure scenarios are provided in Appendix A: NSS-HA Recipe

Server response to a failure

The server response to a failure event within the HA cluster was recorded. Time to recover from a failure was used as a performance metric. Time was measured from the point when the fault was injected in the server running the HA service (active) until the service was migrated and running on the other server (passive).

1) Server failure - simulated by introducing a kernel panic.
   When the active server fails, the heartbeat between the two servers is interrupted. The passive server waits for a defined period of time and then attempts to fence the active server. Once fencing is successful, the passive server takes ownership of the cluster service. Clients cannot access the data until the failover process is complete.

2) Heartbeat link failure - simulated by disconnecting the private network link on the active server.
   When the heartbeat link is removed from the active server, both servers detect the missing heartbeat and attempt to fence each other. The active server is unable to fence the passive since the missing link prevents it from communicating over the private network. The passive server successfully fences the active server and takes ownership of the HA service.

3) Public link failure - simulated by disconnecting the InfiniBand or 10 Gigabit Ethernet link on the active server.
   The HA service is configured to monitor this link. When the public network link is disconnected on the active server, the cluster service stops on the active server and is relocated to the passive server.

4) Private switch failure - simulated by powering off the private network switch.

   When the private switch fails, both servers detect the missing heartbeat from the other server and attempt to fence each other. Fencing is unsuccessful since the network is unavailable and the HA service continues to run on the active server.

5) Fence device failure - simulated by disconnecting the iDRAC cable from the server.
   If the iDRAC on a server fails, the server will be fenced via the network PDUs which are defined as secondary fence devices in the cluster configuration files.

6) One SAS link failure - simulated by disconnecting one SAS link between the PowerEdge R710 server and the PowerVault MD3200 storage.
   In the case where only one SAS link fails, the cluster service is not interrupted. Since there are multiple paths from the server to the storage, a single SAS link failure does not break the data path from the clients to the storage and thus does not trigger a cluster service failover.

   For cases (1) through (6) it was observed that the HA service failover takes in the range of a half to one minute. This reaction time is faster with this version of the cluster suite than with the previous version [4]. Thus in a healthy cluster, any failure event should be noted by the Red Hat cluster management daemon and acted upon within minutes. Note that this is the failover time on the NFS servers; the impact to the clients could be longer.

7) Multiple SAS link failures - simulated by disconnecting all SAS links between one PowerEdge R710 server and the PowerVault MD3200 storage.
   When all SAS links on the active server fail, the multipath daemon on the active server retries the path to the storage based on the parameters configured in the multipath.conf file. This is set to 150 seconds by default. After this process times out, the HA service will attempt to failover to the passive server.

   If the cluster service is unable to cleanly stop the LVM and the file system because of the broken path, a watchdog script will reboot the active server after five minutes. At this point the passive server will fence the active server, restart the HA service and provide the data path to the clients. This failover can therefore take anywhere in the range of three to eight minutes.

Impact to clients

Clients mount the NFS file system exported by the server using the HA service IP. This IP is associated with either an InfiniBand or a 10 Gigabit Ethernet network interface on the NFS server. To measure any impact on the client, the dd utility and the iozone benchmark were used to read and write large files between the client and the file system. Component failures were introduced on the server while the client was actively reading and writing data from the file system.

In all scenarios it was observed that the client processes complete the read and write operations successfully. As expected, the client processes take longer to complete if the process is actively accessing data during a failover event. During the failover period when the data share is temporarily unavailable, the client process was observed to be in an uninterruptible sleep state.

Depending on the characteristics of the client process it can be expected to abort or sleep while the NFS share is temporarily unavailable during the failover process. Any data that has already been written to the file system will be available. The cluster configuration includes several design choices to protect data during a failover scenario as describe in Section 5.3.

For read and write operations during the failover case, data correctness was successfully verified using the checkstream utility.

Details on the tools used are provided in Appendix B: Benchmarks and test tools.

6. Performance tests

This section presents the results of the performance related tests conducted on the NSS-HA solution. The method used for performance analysis is similar to previous versions of the solution and is described in detail here. The larger capacities and updated performance characteristics of this version of the solution are some of the big differentiators for this release of the solution.

All performance tests were done in a failure free scenario to measure the maximum capability of the solution. Analysis focused on three types of IO patterns: large sequential reads and writes, small random reads and writes, and metadata operation related tests.

Both the 10 Gigabit Ethernet and the IP-over-InfiniBand (IPoIB) cases were benchmarked. The 64-node compute cluster described in Section 5.2 was used to provide IO load to the NSS-HA solution. Each test was run over a range of clients to test the scalability of the solution.

Both the 144TB and 288TB configurations were benchmarked. The 288TB solution provides double the capacity and utilizes twice the disk spindles as the 144TB solution as describes in Section 4.1, Table 2. Recall from Section 4.2 that a 288TB configuration has two 144TB portions concatenated together. For tests on the 288TB configuration, care was taken to exercise both parts of the file system by ensuring multiple concurrent client streams access different parts of the file system.

The iozone and mdtest utilities were used in this study. Details on the benchmarks are provided in Appendix B: Benchmarks and test tools.

Iozone was used for the sequential and random tests. For sequential tests, a request size of 1024KB was used. The total amount of data transferred was 128GB to ensure that the NFS server cache was saturated. Random tests used a 4KB request size and each client read and wrote a 2GB file.

Metadata tests were performed using the mdtest benchmark and include file stat, create and delete operations.

While these benchmarks do not cover every I/O pattern, they help characterize the I/O performance of the NSS-HA solution.

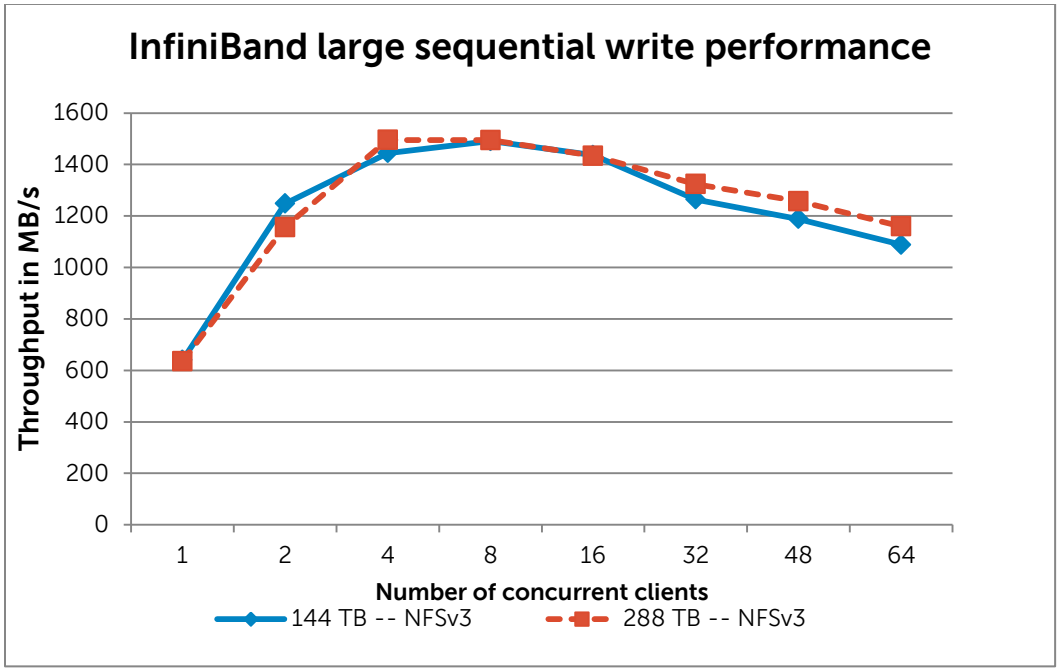
As mentioned in Section 5.3 bullet (12), performance was evaluated using NFSv3 as well as NFSv4. In the following section performance results are presented on NFSv3 and a comparison of NFSv3 and NFSv4 is presented section 6.5.

6.1. InfiniBand sequential writes and reads

This section presents the results of sequential IO tests over the InfiniBand on the 144TB and 288TB configurations. The IP-over-IB driver supports two modes of operation – reliable connected mode and unreliable datagram mode. The two modes differ mainly in packets delivery reliability and in the size of the MTU. This can be changed at runtime by managing the contents of the `/sys/class/net/ib0/mode` file. Results in this section are using the default connected mode. As a follow on to this document, a blog post on www.hpcatdell.com will discuss the performance of the datagram mode.

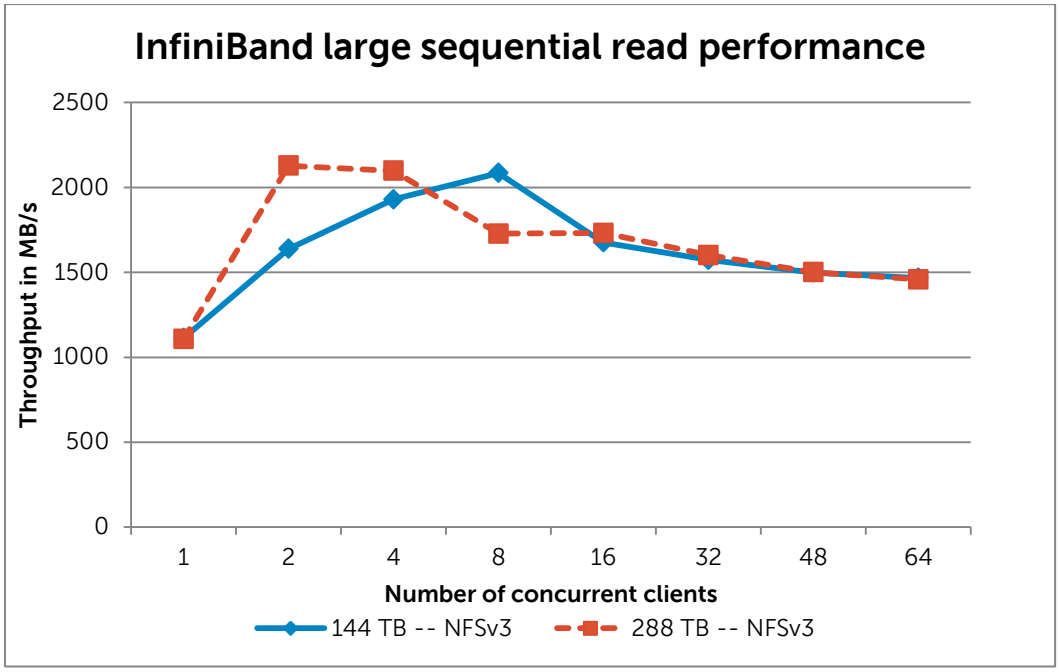
From Figure 8 it is seen that the peak throughput for sequential writes is ~1500MB/s. The write throughput is constrained by the NSS-HA design choices of RAID controller write cache mirroring and the NFS sync export option. Thus the performance of the 144TB and 288TB configurations is similar.

Figure 8. InfiniBand large sequential write performance



The sequential read performance as shown in Figure 9 peaks at ~2000MB/s.

Figure 9. InfiniBand large sequential read performance



6.2. 10 Gigabit Ethernet sequential writes and reads

Results of sequential write and read tests over the 10 Gigabit Ethernet link to the NFS server are shown in Figure 10 and Figure 11. From the graphs it can be seen that the NSS-HA solution scales well for both writes and reads. The peak throughput saturates around 1200MB/s which reaches the maximum capability of the public network that connects the client cluster to the server. Both the 144TB and 288TB configurations exhibit similar performance trends since the network is the limiting factor in this case and not the number of disks in the solution. Note that for these tests, each client is connected to the NFS server via a single Gigabit Ethernet link.

Figure 10. 10GbE large sequential write performance

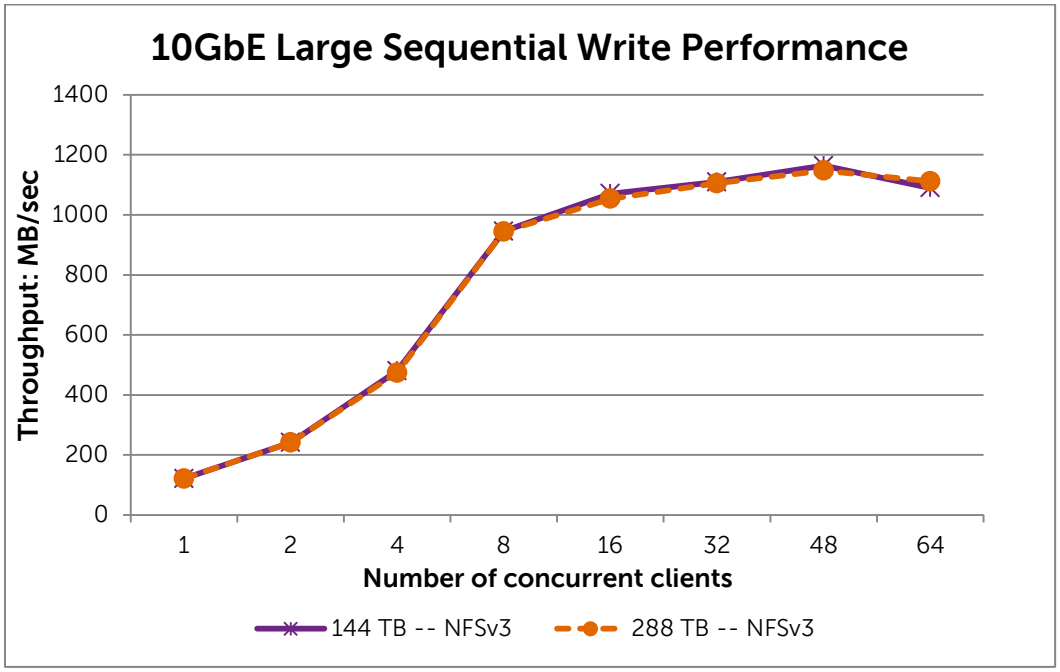
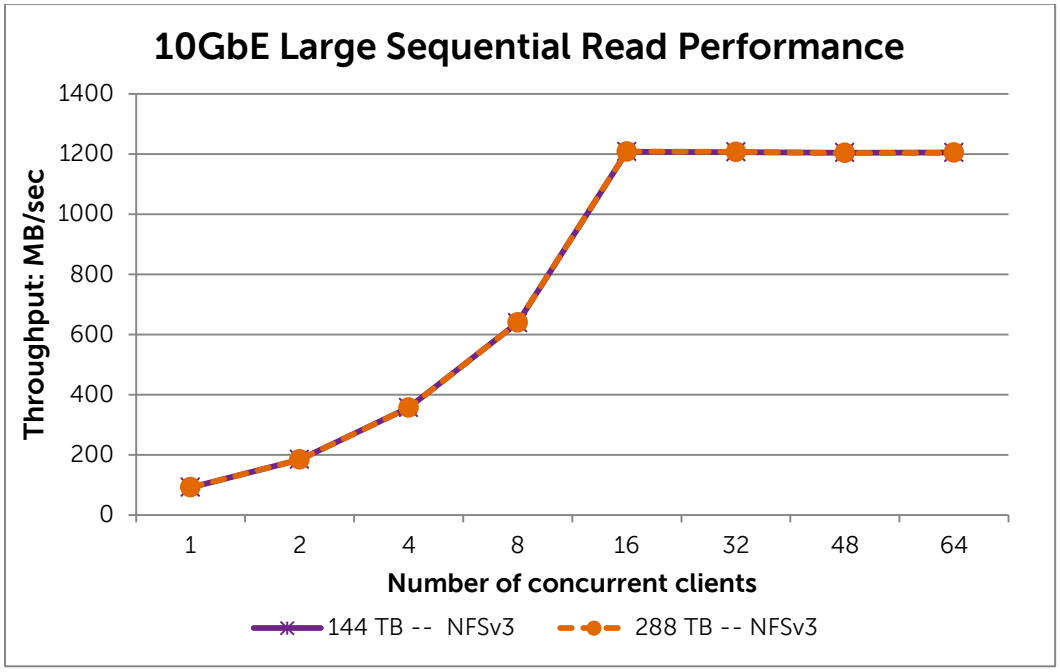
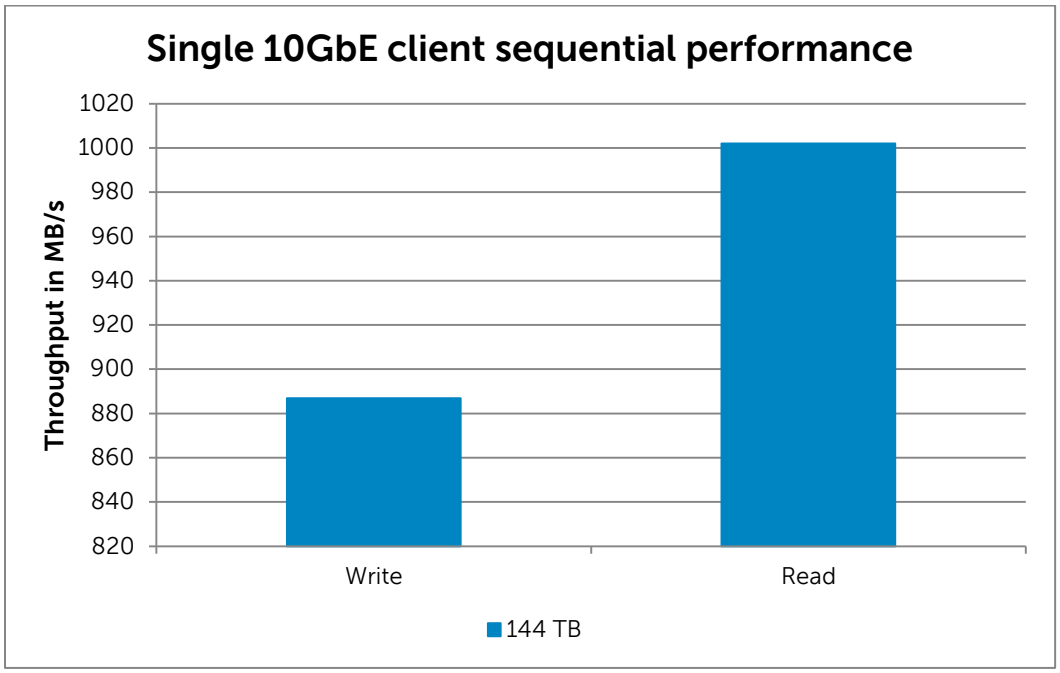


Figure 11. 10GbE large sequential read performance



A single NFS client with a 10 Gigabit Ethernet to the storage solution was also tested. In this case the single 10GbE client and the NFS server were both directly connected to the PowerConnect 8024 10GbE switch. Results for this test using the NFSv3 protocol are shown in Figure 12. From the graph it is seen that write throughput is ~880MB/s and read throughput is ~1000MB/s for 144TB configuration. The 288 TB configuration is similar in performance.

Figure 12. Single 10gbE client sequential performance

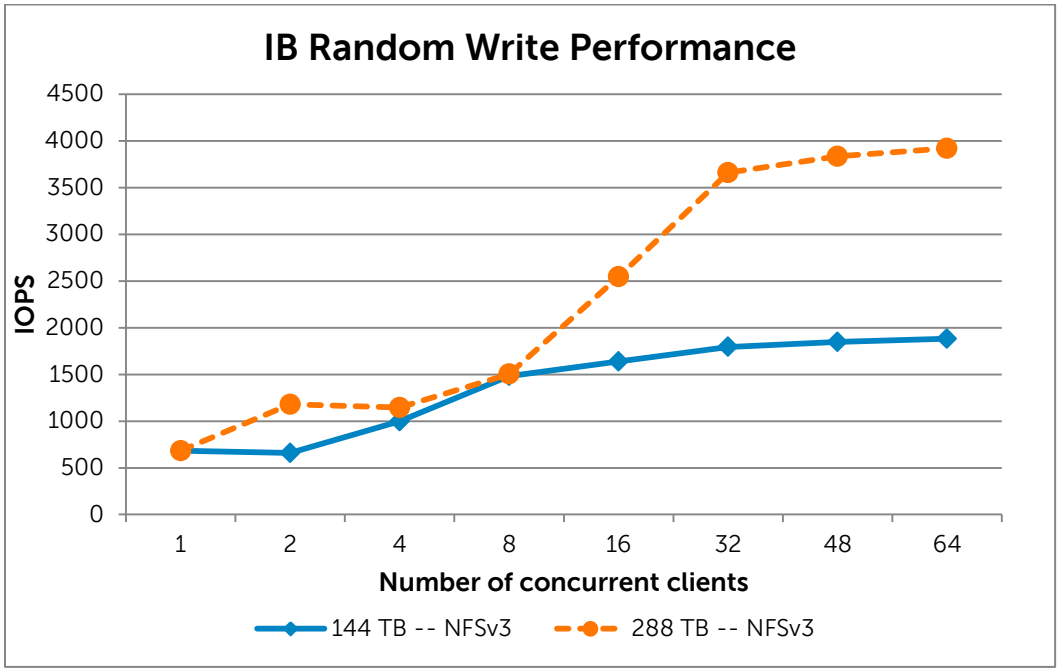


6.3. Random writes and reads

This section presents the results for random write and read tests over IPoIB. From past experience, random test results were expected to be very similar for 10GbE and IPoIB; however that was not observed in this environment. The general pattern and curve of the graphs remained the same for both networks, but the absolute performance varied depending on the number of concurrent clients in the tests. Results for 10GbE will be presented in a blog post on www.hpcatdell.com at a later date.

Figure 13 presents the results of the random write tests. The figure shows the aggregate IO when a number of clients are simultaneously writing to the storage. Peak IOPS (IO operations per second) for the 144TB solution are close to 2000 IOPS. The 288TB solution has double the number of disk drives when compared to the 144TB solution and the peak performance is about double at 4000 IOPs, showing the scalability of the NSS-HA solution.

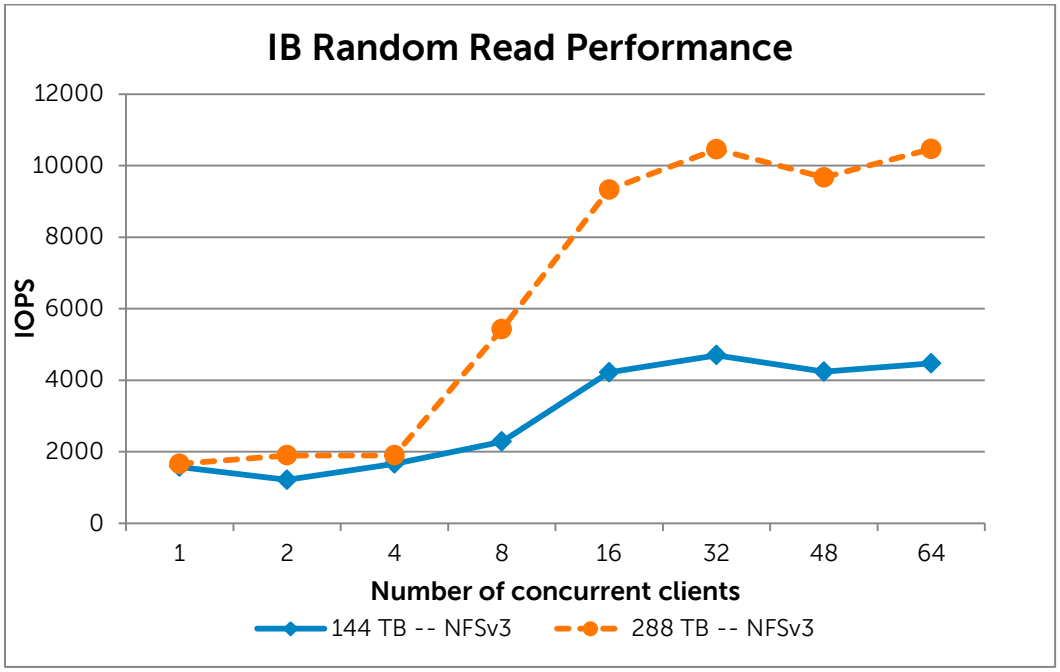
Figure 13. InfiniBand random write performance



The NSS-HA write performance is limited by several design factors including write cache mirroring on the RAID controllers, XFS wsync mount option and NFS sync export option.

The read performance is not impacted by these choices, additionally, read operations have a lower overheard than write operations in RAID 6. For these reasons read performance is much better than writes. Results of the random read tests are shown in Figure 14. From the figure, random read IOPs scale to ~10000 IOPS for the 288 TB configuration and to ~4500 IOPS for the 144 TB configuration. Both random writes and reads are impacted by disk seek latency and the additional disk spindles in the 288 TB solution help the larger capacity configuration perform better. Actual workloads could see better read IOPS than presented here for situations where cache on the client and NFS server can be exploited. As part of this study (but not shown here), read performance of the order of ~50000 IOPS was measured for the cases where cache helped performance

Figure 14. InfiniBand random read performance



6.4. Metadata tests

From past experience, it was expected that metadata test results would be very similar for 10GbE and IPoIB; however that was not observed in this environment. For both networks the general curve of the graphs remained the same, but the absolute performance varied depending on the number of concurrent clients in the tests. For some client counts IPoIB performed better than 10GbE, but for others, 10GbE performed better than IPoIB, and the variation was observed to be statistically significant.

This release of the solution includes updated firmware on the storage arrays, larger capacity configurations, a newer version of the operating system and XFS file system. With this release of the solution, metadata performance is much higher than in the previous version showing that both IPoIB and 10GbE are able to scale. However, for the cases where 10GbE performance is better than IPoIB, it is possible that the solution is now stressing the IPoIB implementation more than in previous versions and thus exposing some issues. That being said, the metadata performance of the solution is impressive and is expected to meet requirements of most clusters. Results for 10GbE will be presented in a blog post on www.hpcatdell.com at a later date. The InfiniBand results are presented and discussed in this section.

Figure 15, Figure 16 and Figure 17 show the results of file create, stat and remove operations, respectively. Recall that the HPC compute cluster has 64 compute nodes. In the graphs below each client executed a maximum of one thread for client counts up to 64. For client counts of 128, 256 and 512, each client executed 2, 3 or 4 simultaneous operations.

Similar performance for the 144TB and 288TB configurations indicates that the number of disk spindles is not a factor. The results demonstrate the scalability of the NSS-HA solution for such operations. As

expected the file create and file remove performance is similar, since both involve write-type operations.

Figure 15. InfiniBand file create performance

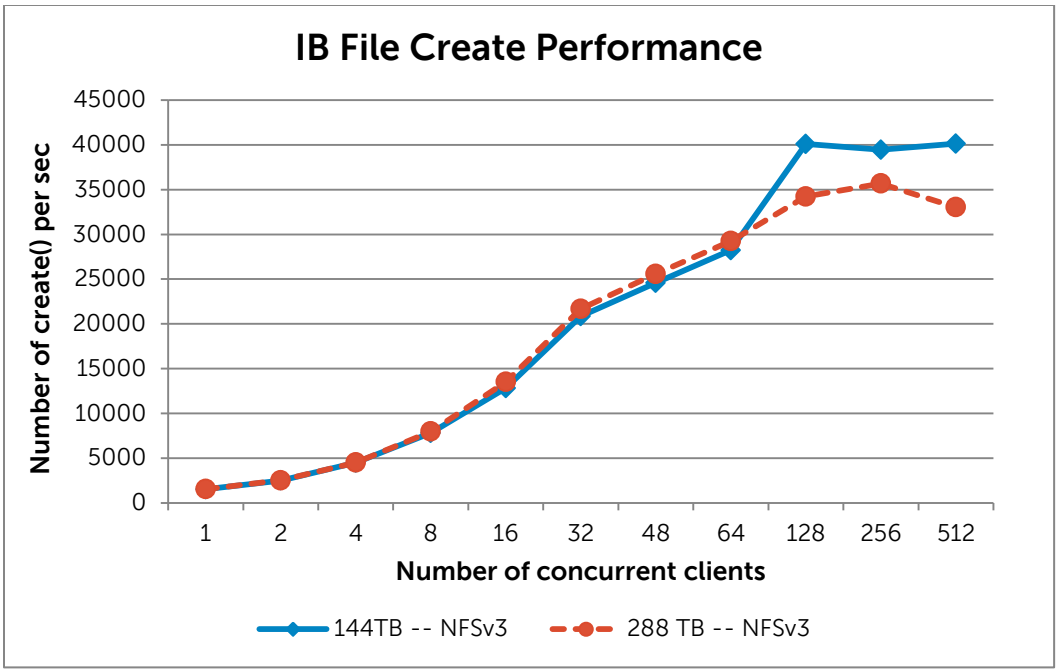


Figure 16. InfiniBand file stat performance

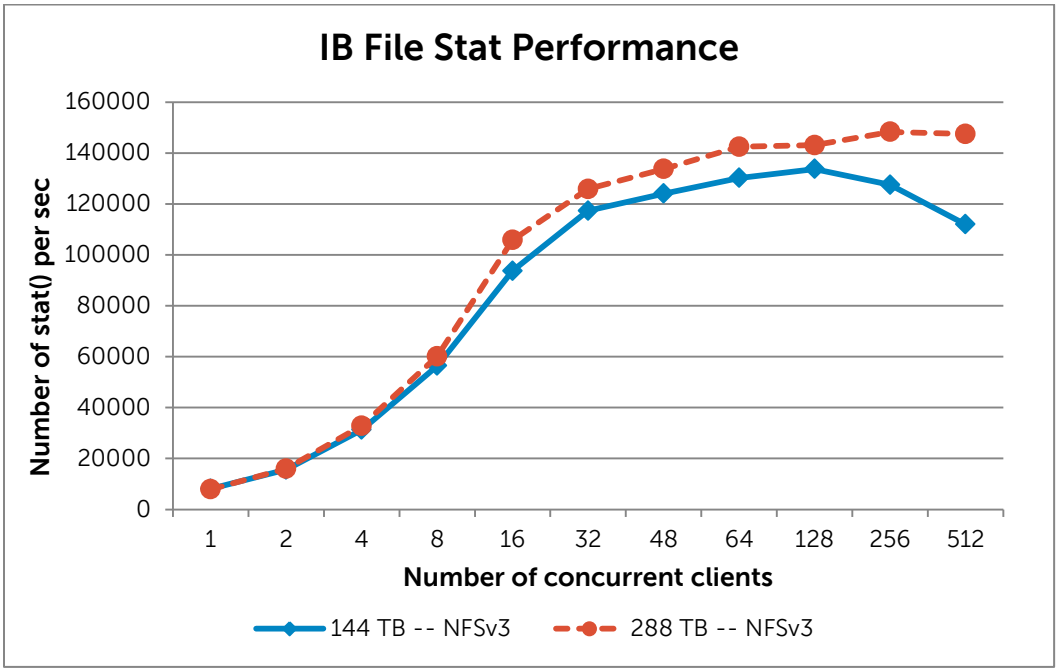
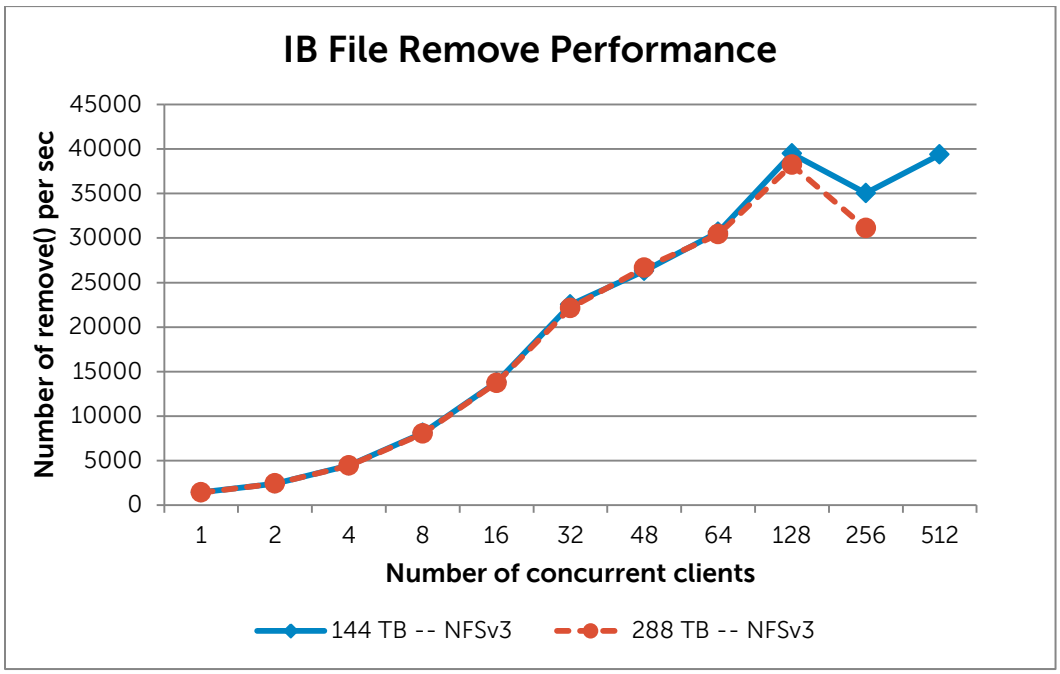


Figure 17. InfiniBand file remove performance



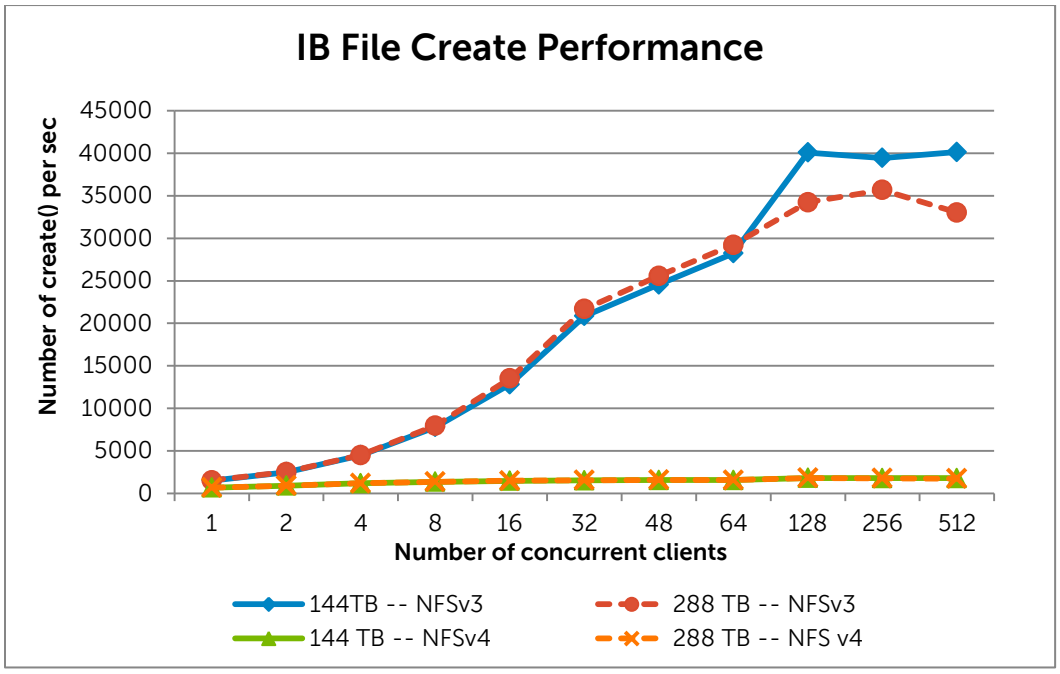
6.5. NFSv3 compared to NFSv4

During the design and analysis of the NSS-HA solution it was found that NFSv3 provides better performance than NFSv4 for certain scenarios. This section describes the deltas in performance between NFSv3 and NFSv4. In certain situations the security enhancements of NFSv4 might be more important than any loss in performance. The results presented in this section help quantify the performance loss and pinpoint the scenarios where a difference in performance can be expected.

For the three types of IO workloads tested in this study, it was found that the most dramatic difference between NFSv3 and NFSv4 performance was in the metadata file create tests. These results are presented later in this section in Figure 18. Metadata file stat were measured to be within 15% of each other for NFSv3 and NFSv4; NFSv3 performed better for low client counts and NFSv4 performed better for the higher client counts. Metadata remove operations were 10 – 28% faster with NFSv3.

Figure 18 compares NFSv3 and NFSv4 performance on the metadata file create tests. As seen from the graph, the 144TB and 288TB configurations perform the same with either protocol version, i.e., the capacity of the configuration does not matter, the NFS protocol version is the main differentiator. NFSv3 performance is an order of magnitude better than NFSv4. NFSv4 file creates saturate at about 1550 create operations per second. With NFSv3, the peak performance is close to 34000 create operations per second. With the current version of NFSv4 the state locking implementation causes the file create operations to be serialized, hindering performance [(5)]. This issue is expected to be alleviated in future versions.

Figure 18. InfiniBand NFSv3 and NFSv4 file create performance



With IPoIB, sequential reads were found to be up to 22% better with NFSv4 while sequential write throughput is was up to 22% worse with NFSv4.This is shown in Figure 19 for a 288TB configuration. The 144TB configuration shows a similar trend.

Figure 19. InfiniBand NFSv3 and NFSv4 sequential performance

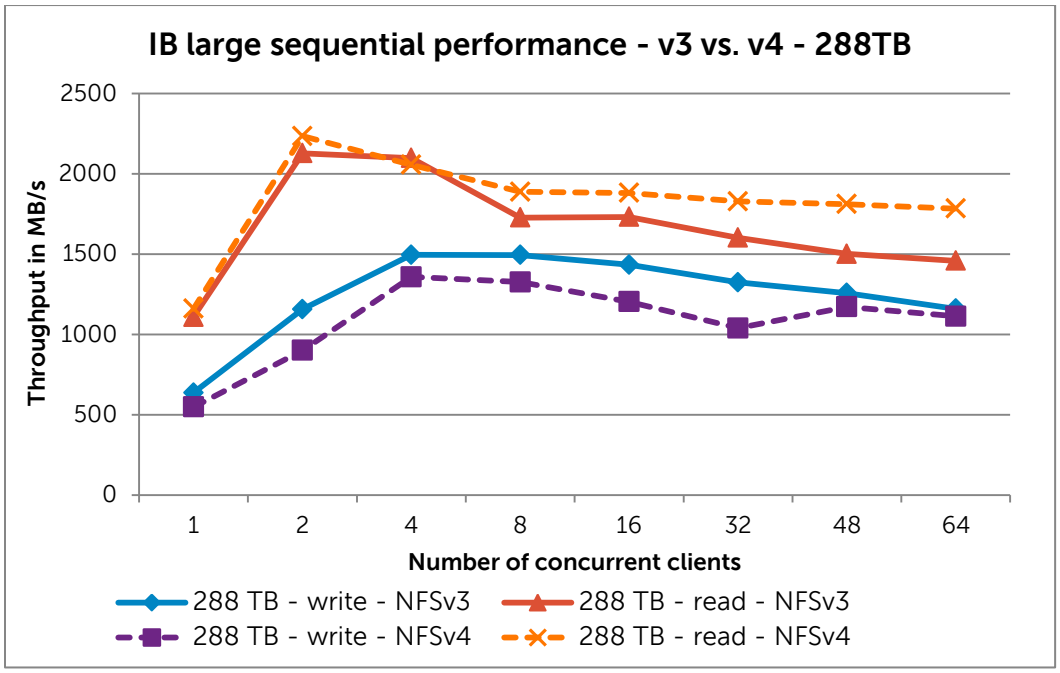


Figure 20. InfiniBand NFSv3 and NFSv4 random performance

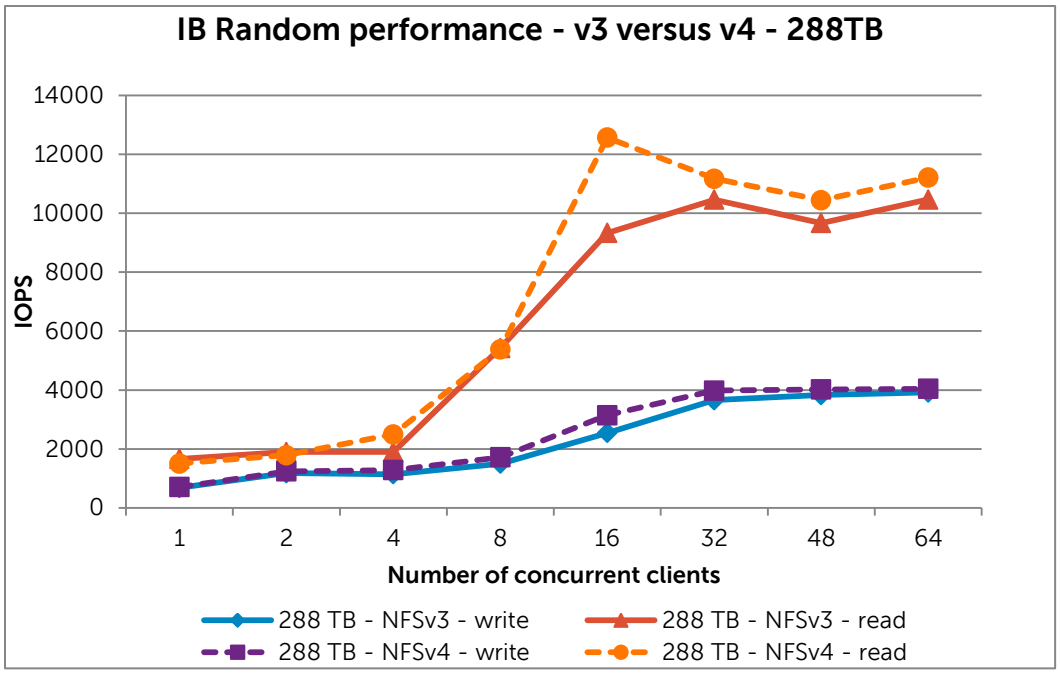


Figure 20 shows the difference in random performance between NFSv3 and NFSv4 for a 288TB configuration. For IPoIB random write operations, NFSv4 was better by 5% to 23%. On random read

operations, NFSv3 versus NFsv4 performance depended on the number of concurrent clients in the test. For single client runs NFSv3 was better by 9%. With 4 and 16 clients NFSv4 was substantially better than NFSv3 by up to 35%. For all other cases, NFSv4 was marginally better. The 144TB configuration showed a similar trend.

Due to time constraints, 10GbE performance with NFSv4 was not measured in time for this publication. This portion of the study will be released as a blog on www.hpcatdell.com at a later date.

7. Conclusion

This solution guide provides details on Dell's latest NFS Storage Solution for HPC. With this release, the Dell NSS solutions grow to support XFS-based file systems at capacities larger than 100TB. The Dell NFS Storage Solution is available with deployment services and full hardware and software support from Dell. This document provides complete technical details on architecture, design, performance analysis and configuration of such solutions.

8. References

1) XFS: A high-performance journaling file system.
http://oss.sgi.com/projects/xfs/

2) Quick SAS Cabling Guide --A Dell Technical White Paper.
http://www.dell.com/downloads/global/products/pvaul/en/powervault-md3200-m3200i-cabling-guide.pdf

3) Red Hat Enterprise Linux 6 Cluster Administration -- Configuring and Managing the High Availability Add-On.
http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Cluster_Administration/Red_Hat_Enterprise_Linux-6-Cluster_Administration-en-US.pdf

4) Dell HPC NFS Storage Solution High Availability Configurations, Version 1.1
http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/dell-hpc-nssha-sg.pdf

5) Bugzilla - NFSv4 server file creates are slow
https://bugzilla.redhat.com/show_bug.cgi?id=771776

Appendix A: NSS-HA Recipe (Updated May 2012)

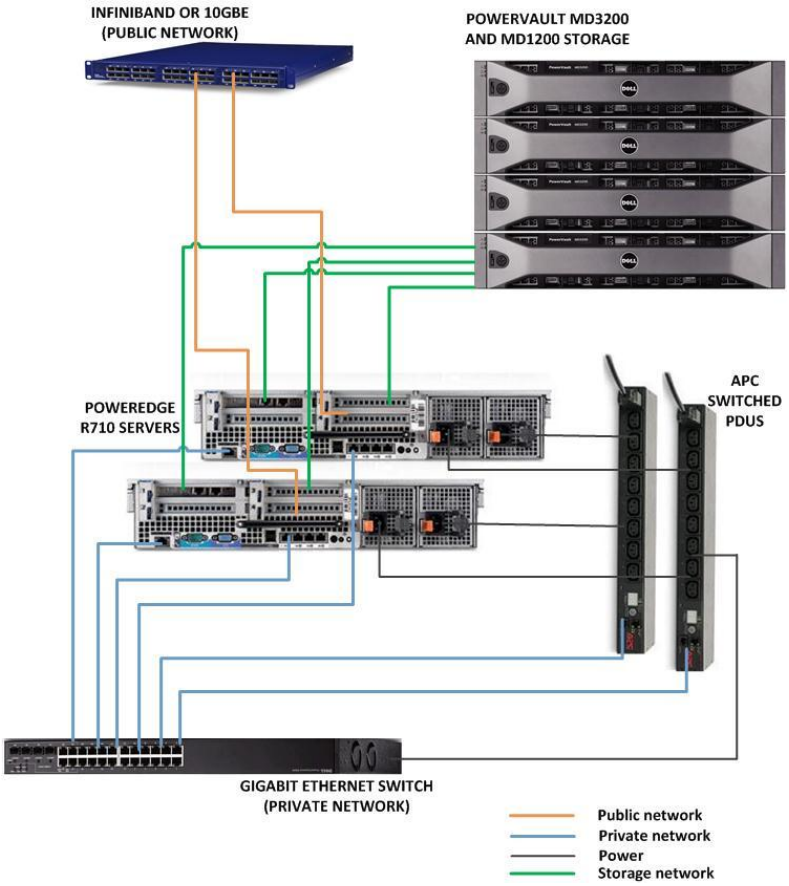
Contents

A.1. Pre-install preparation 39
A.1.1. NSS-HA cluster specification 40
A.1.2. Checklist 42
A.2. Server hardware setup 43
A.2.1. Checklist 44
A.3. Server software configuration 45
A.3.1. Install RHEL 6.1, configure swap disks, and install XFS packages. 45
A.3.2. Configure Multipath 46
A.3.3. Install Mellanox OFED package and set network IPs 48
A.3.4. Install Operating system and storage management tools 49
A.3.5. Network security setting 50
A.3.6. Configure startup services. 51
A.3.7. Configure fence devices 51
A.3.8. Checklist 53
A.4. Performance tuning on the server 54
A.4.1. Checklist 55
A.5. Storage hardware setup 56
A.6. Storage configuration 59
A.6.1. Checklist 61
A.7. NSS HA cluster setup 62
A.7.1. Prepare 62
A.7.2. Create HA cluster 63
A.7.3. Verification 65
A.8. Quick test of HA setup 68
A.9. Useful commands and references 69
A.9.1. Manually modify cluster configuration file 69
A.9.2. Manually stop, disable, start and relocate the cluster service 69
A.9.3. Debug HA cluster configuration 70
A.9.4. Remove a node from HA cluster 70
A.9.5. Configure the shared storage array manually 71
A.10. Performance tuning on clients 74
A.11. Scripts 75

A.1. Pre-install preparation

The following figure shows an NSS HA cluster. The equipment list for NSS-HA includes:

- 2 PowerEdge R710 servers
- 1 PowerConnect 5424 Gigabit Ethernet switch
- 2 Switched APC AP7921 power PDUs
- 1 PowerVault MD3200
- 3 PowerVault MD1200s for 144TB configuration, 7 PowerVault MD1200s for 288 TB configuration



A.1.1. NSS-HA cluster specification

The section gives a detailed specification for constructing an NSS-HA solution, including:

- IP addresses and hostnames for the two R710 servers
- 10GbE / IPoIB IP addresses for the two R710 servers
- iDRAC IP addresses, login names, and passwords
- APC PDU IP addresses, login names, passwords, and ports used by the two R710 servers
- Logical volume names, volume group names, and XFS file system mount points
- NFS IP addresses used to mount NFS
- Scripts for cluster

The configuration specified in this section will be used later in the recipe to construct the HA-cluster. Any deviation from this configuration should be noted carefully since it require corresponding changes in the cluster configuration script file later in the recipe. This section is provided as a reference of all the configuration details.

Gigabit Ethernet IP addresses for the private network between the servers and the PDUs.

R710 (hostname: active)	NIC port 1 IP: 15.15.10.1/24
R710 (hostname: passive)	NIC port 1 IP: 15.15.10.2/24
iDRAC (installed on active)	15.15.10.201/24
iDRAC (installed on passive)	15.15.10.202/24
PDU 1	15.15.10.101/24
PDU 2	15.15.10.102/24

Note: All IPs should be on the same subnet.

iDRAC and APC PDU settings.

iDRAC (installed on active)	Login name: root password: calvin
iDRAC (installed on passive)	Login name: root password: calvin
PDU 1	Login name: apc password: apc port 2 for active, port 3 for passive

PDU 2	Login name: apc password: apc port 2 for active, port 3 for passive

Note: Instructions will be provided in section A.3.7 to configure iDRAC and APC PDU.

Logical volume names, volume group names, and XFS file system mount point

Logical volume	LVMD1
Volume group names	VGMD1
XFS mount point	mount point: /mnt/xfs1 mount option: `noatime,allocsize=1g,nobarrier,inode64,logbsize=262144,wsync`

Note: Please do not make any change to the XFS mount option without the guideline from DELL representatives; it may incur unexpected performance and reliability issues.

Public network IP addresses and NFS mounting IP.

10GbE/IPoIB IP address (active)	10.10.10.201/16
10GbE/IPoIB IP address (passive)	10.10.10.202/16
NFS IP for clients to mount NFS	NFS IP: 10.10.10.200 NFS export option: `rw,sync,no_root_squash`

Note: These IPs should be on the same subnet as the clients. Please do not make any change to the NFS export option without the guideline from DELL representatives; it may incur unexpected performance and reliability issues.

Scripts for cluster configuration and monitoring

Script name	**usage**
config_cluster.sh	Generating cluster configuration file

nssha61_single.py	Configuring storage devices, including creating/removing PVs, VGs, LVs, and XFS file system.
sas_path_check.sh	Used by HA cluster management tool to monitor the status of SAS paths.
ibstat_script.sh	If IPoIB is deployed, the script is used by HA cluster management tool to monitor IB link status.

Note: all scripts are attached with this document and can be found in section A.11

Very important: Once the cluster configuration file in section A.7.2 is generated, please double check the configuration file to make sure all information is correct according to the specification.

Note: All settings above were used in the test configuration; they can be changed according to the actual configuration. Section A.9.1 will provide the instructions to manually change them during cluster construction.

A.1.2. Checklist

Before moving to the next section, please make sure all following items are ready.

Items	Notes
✓ Two PowerEdge R710s	Section 5.2, Table 5 lists the detail configuration.
✓ One PowerConnect 5424 Gigabit Ethernet switch	
✓ Two Switched APC AP7921 power PDUs	Check https://access.redhat.com/knowledge/articles/28603 for other supported APC PDU models.
✓ One PowerVault MD3200	
✓ Three PowerVault MD1200s for 144TB configuration, or seven PowerVault MD1200s for 288 TB configuration	Section 5.2, Table 5 lists the detail configuration.
✓ NSS-HA cluster specification	Prepare a specification according to section A.1.1.

A.2. Server hardware setup

1. Prepare two PowerEdge R710 servers (called “active” and “passive”). Configure each server as follows.

   - One PERC H700 and 5 local disks each of 146 GB.
   - Configure 2 disks in RAID 1 with 1 additional disk designated as the hot spare. This will be used for the operating system.
   - Configure 2 disks in RAID 0, this will be used as swap.
   - 10 Gigabit Ethernet card OR InfiniBand card in slot 4, a PCI-E x8 slot.
   - Two SAS 6 Gbps cards, one in slot 1, a PCI-E x4 slot and one in slot 3, a PCI-E x8 slot. The two cards will connect to the MD3200 storage.
   - iDRAC enterprise.
   - Dual power supplies.

   A reference for the PERC H700 storage controller is provided below. The PowerEdge R710 server might ship from the Dell factory with 2 disks already configured in RAID 1. Insert the additional 3 disks into the server and use the referenced below to add a disk as a hot spare for the RAID 1 setup and to configure the remaining two disks in RAID 0.

   Reference: Dell PowerEdge RAID Controller (PERC) H700 and H800 User’s Guide, http://support.dell.com/support/edocs/storage/Storlink/H700H800/en/UG/PDF/H700H800.pdf

2. Set up a Gigabit Ethernet switch for the private network. Power the Gigabit switch from one of the APU Power PDUs. You will need at least 6 Ethernet ports on the switch - one port each for:

   - Ethernet cable from power PDU1
   - Ethernet cable from power PDU2
   - iDRAC enterprise from active server
   - iDRAC enterprise from passive server
   - NIC1 from active server
   - NIC1 from passive server

3. Connect the Ethernet port of each PDU to the Gigabit Ethernet private switch.

4. For each R710, cable the onboard NIC1 and iDRAC enterprise port to the Gigabit Ethernet private switch.

5. Set up two switched power PDUs with at least 3 power ports each.

   - PDU 1, port 1 for the Gigabit switch for the private network.
   - PDU 1, port 2 for power supply 1 on active
   - PDU 1, port 3 for power supply 1 on passive
   - PDU 2, port 2 for power supply 2 on active
   - PDU 2, port 3 for power supply 2 on passive

6. For each R710, plug in the two power supplies on the server to each of the two switched power PDUs as per the designated port specification in section A.1.1, and power on the two PDUs.

7. Connect the IB or 10GbE network to the “public” network. This is the network the NFS clients are connected to.

A.2.1. Checklist

Before moving to the next section, please make sure all following tasks are completed.

Tasks	Notes
✓ Install hard disks, SAS cards, 10gbE card or IB card, and iDRAC enterprise on each R710.	Section A.2, step 1.
✓ Configure local disks on each R710.	Section A.2, step 1.
✓ Connect all PDUs, iDRACs, and two R710s to a Gigabit switch.	Section A.2, step 2, step 3, step 4.
✓ Connect the switch and two R710s to the two PDUs.	Section A.2, step 5 and 6. Make sure that each R710 are connected to the designated ports.
✓ Connect each R710 to the public network via 10gbE or IB links.	Section A.2, step 7.

A.3. Server software configuration

All the operations below apply to each R710.

A.3.1. Install RHEL 6.1, configure swap disks, and install XFS packages.

1. Install the RHEL6.1 x86_64 operating system (kernel version 2.6.32-131.0.15.el6.x86_64) on the RAID1 virtual disk.
   - Make sure MD storage is not attached to the servers during the OS installation.
   - Do not select or include the RAID0 virtual disk for use in the OS install, it will be used for swap space in steps below.
2. After the OS is installed, setup swap on the RAID0 device.
   - `# parted -l` will show the device associated with the PERC H700 with a size appropriate for the RAID 0, about 292 GiB; such as `/dev/sdb:`

     ```
     ...
     Model: DELL PERC H700 (scsi)
     Disk /dev/sdb: 292GB
     ...
     ```
   - `# mkswap -L SWAP /dev/sdb`
   - `# swapon -p 10 /dev/sdb`
   - Edit /etc/fstab and add an entry for this swap device. Please make sure that the entry is listed BEFORE the default swap space that was created during OS install, if any.

     "`LABEL=SWAP swap  swap  pri=10      0   0`"

     Alternatively, the UUID can be used for identifying the new swap space:

     ```
     # blkid | grep swap | grep -v lv
     /dev/sdb1: UUID="c9d4a87c-22af-4596-97ee-511a67ddff13" TYPE="swap"
     ```

     In that case, the entry in /etc/fstab will look like:

     "`UUID=c9d4a87c-22af-4596-97ee-511a67ddff13  swap  swap  pri-10  0 0`"
   - Test that swap can be enabled automatically when the server boots.

     ```
     # swapoff -a
     # swapon -s
     # swapon -a
     # swapon -s
     ```
3. Obtain the XFS packages from the Red Hat Network (http://rhn.redhat.com) and install them.

   ```
   xfsprogs-3.1.1-4.el6.x86_64.rpm
   xfsprogs-devel-3.1.1-4.el6.x86_64.rpm
   xfsdump-3.0.4-2.el6.x86_64.rpm
   ```

A.3.2. Configure Multipath

On each R710, exclude the local disks from the control of `multipathd`.

1. Make sure the `multipath` software is installed. To verify if the multipath software is installed:

   ```
   # rpm –a | grep multipath
   device-mapper-multipath-0.4.9-41.el6.x86_64
   device-mapper-multipath-libs-0.4.9-41.el6.x86_64
   ```

   If the packages are not present, install them and then run the following command to create the `/etc/multipath.conf` file:

   ```
   #/sbin/mpathconf --user_friendly_names y
   ```

   Alternatively, copy the example included in the multipath package to `/etc/multipath.conf`:

   ```
   # cp –p /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf
   /etc/multipath.conf
   ```

2. Make sure the service `multipathd` is not started.

   ```
   #service multipathd status
   Multipathd is stopped
   ```

   Otherwise, local disks will be controlled by the `multipathd` service and that will prevent stopping or restarting the service, with the following message:

   ```
   Root is on a multipathed device, multipathd cannot be stopped
   ```

3. Blacklist local devices to prevent problems when installing MDSM later in section A.3.4.

   a. Identify the local virtual disks created in section A.2.

   ```
   # parted -l 2>&1 |grep -A1 -B1 'Model: DELL PERC H700'
   Model: DELL PERC H700 (scsi)
   Disk /dev/sda: 146GB
   --

   Model: DELL PERC H700 (scsi)
   Disk /dev/sdb: 292GB
   ```

   Note: Virtual disks (RAID 0 and 1) devices are listed as PERC devices, hard disks not part of a RAID device will be shown under the hard disk manufacturer and model number, so the above command will not list them. If that is the case, make sure multipathd is not enabled (`chkconfig multipathd off`), reboot and enter the PERC H700 controller BIOS to create the virtual disks, reboot again and continue from the previous step.

   b. Get the WWID of those local devices.

   ```
   # /sbin/scsi_id --whitelisted --device=/dev/sda
   36842b2b0723e980017184fb50ae00d6a
   # /sbin/scsi_id --whitelisted --device=/dev/sdb
   36842b2b0723e980017184fcf0c6c56e9
   ```

Then, edit the file `/etc/multipath.conf`, search for the `blacklist` section and uncomment it or modify it if already uncommented. Add the "wwid" entries for the local virtual disks to make it look like the following example:

```
blacklist {
        wwid    "36842b2b0723e980017184fb50ae00d6a"
        wwid    "36842b2b0723e980017184fcf0c6c56e9"

…

{Rest of your original blacklist, if any}

}
```

c. Remove any local disks entries from the file `/etc/multipath/bindings`.

Note: If multipathd was never started (i.e., if it was just installed in this section), this file will not exist, and skip the step.

```
# sed -i 's/^.*36842b2b0723e980017184fb50ae00d6a.*$//'
/etc/multipath/bindings
# sed -i 's/^.*36842b2b0723e980017184fcf0c6c56e9.*$//'
/etc/multipath/bindings
```

4. Finally, start the service (or start it, if you just installed it), enable it to start upon reboots and check if your local disks are now blacklisted with the commands:

```
# service multipathd restart
Stopping multipathd daemon:                                [  OK  ]
Starting multipathd daemon:                                [  OK  ]


# chkconfig multipathd on
```

5. Verify that the local disks are blacklisted.

```
# multipath -v3 | grep blacklisted | grep "PERC H700"
Feb 02 18:53:19 | sda: (DELL:PERC H700) wwid blacklisted
Feb 02 18:53:19 | sdb: (DELL:PERC H700) wwid blacklisted
```

If the command to restart the service fails or the local disks are not blacklisted, then reboot the server, and check again if they are blacklisted when the server is up.

Reference: Red Hat Enterprise Linux 6 DM Multipath: DM Multipath Configuration and Administration; Edition 1, section 3.2 Ignoring Local Disks when Generating Multipath Devices. http://www.redhat.com/resourcelibrary/datasheets/red-hat-enterprise-linux-6-dm-multipath-datasheet

A.3.3. Install Mellanox OFED package and set network IPs

1. Install Mellanox OFED 1.5.3-3.0.0 if using InfiniBand (MLNX_OFED_LINUX-1.5.3-3.0.0-rhel6.1-x86_64.iso).

   Note: If 10GbE network is deployed, please skip this step.

   You may need to install the dependencies:

   ```
   glibc-devel-2.12-1.25.el6.i686.rpm
   tcl-8.5.7-6.el6.x86_64.rpm
   tk-8.5.7-5.el6.x86_64.rpm
   ```

2. Configure the IPoIB ib0 address or 10GbE address for the public network as per the specification in Section A.1.1.
3. Set the IP addresses on the private network interface, public network interface and iDRAC NIC on each server according to the NSS-HA cluster specification in section A.1.1

   Make sure that `/etc/hosts` on both servers to contain the entries below for both servers

   ```
   # cat /etc/hosts
   # Do not remove the following line, or various programs
   # that require network functionality will fail.
   127.0.0.1               localhost.localdomain localhost
   15.15.10.1              active.hpc.com  active
   15.15.10.2              passive.hpc.com passive
   ```

   On both servers, disable `NetworkManager` service. HA cluster requires the `NetworkManager` service to be off.

   ```
   # service NetworkManager stop
   # chkconfig NetworkManager off
   ```

4. Set up password-less ssh between the active and passive servers.

   ```
   active# ssh-keygen -t rsa
   active# chmod 700 ~/.ssh
   active# ssh-copy-id –i ~/.ssh/id_rsa.pub passive
   passive# ssh-keygen -t rsa
   passive# chmod 700 ~/.ssh
   passive# ssh-copy-id –i ~/.ssh/id_rsa.pub active
   ```

A.3.4. Install Operating system and storage management tools

1. Install Dell OpenManage Server Administrator (http://downloads.dell.com/sysman/OM-SrvAdmin-Dell-Web-LX-6.5.0-2247.RHEL6.x86_64_A01.5.tar.gz)

   If needed install the included security key using the command:
   ```
   # rpm --import RPM-GPG-KEY
   ```

   If the setup fails citing missing dependencies, install the missing rpms from the RHEL 6.1 DVD

   ```
   libcmpiCppImpl0-2.0.1-5.el6.x86_64.rpm
   sblim-sfcc-2.2.1-4.el6.x86_64.rpm
   sblim-sfcb-1.3.8-1.el6.x86_64.rpm
   openswan-2.6.32-4.el6.x86_64.rpm
   openwsman-server-2.2.3-8.el6.x86_64.rpm
   openwsman-client-2.2.3-8.el6.x86_64.rpm
   ```

   Make sure you select all components for installation and accept starting the services at the end of the installation.

2. Install the MD3200 management software. The console installation is described below, but you can use the GUI installation if preferred (DELL_MDSS_Consolidated_RDVD_3_0_0_18_A00_R314542.iso).

   ftp://ftp.dell.com/RCD/DELL_MDSS_Consolidated_RDVD_3_0_0_18_A00_R314542.iso

   Select the defaults and make sure to do a full installation and select “MD3200 or MD3220” as the storage. Make sure to **NOT REBOOT** at the end of the installation as suggested by the software.

A.3.5. Network security setting

In this step ports will be enabled on both servers. The list of cluster ports to be enabled is in the Red Hat Cluster Administration Guide, section 2.3.

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Cluster_Administration/Red_Hat_Enterprise_Linux-6-Cluster_Administration-en-US.pdf

The list of NFS ports to be allowed in the firewall is listed in this Red Hat document:

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Storage_Administration_Guide/s2-nfs-nfs-firewall-config.html

Launch the firewall configuration tool (system-config-securitylevel), make sure to (select) enable NFSv4 and SSH in the GUI, and exit it.
Then, use the following commands to start the firewall and enable it across reboots, allow portmapper and cluster traffic, and save the changes.

```
# service iptables start; chkconfig iptables on
# iptables -I INPUT -p udp -m state --state NEW -m multiport --dports 111,
2049,5404,5405 -j ACCEPT
# iptables -I INPUT -p tcp -m state --state NEW -m multiport --dports 111,
2049,11111,16851,8084,21064 -j ACCEPT
# service iptables save
```

Note: For NFSv4, unlike NFSv3, all NFS communications are self-contained within port 2049.
If there is no plan to deploy NFSv3, no further configuration is needed for the iptables service. Skip to the next step.

The NFSv3 ports must be allowed through the firewall:

```
# iptables -I INPUT -p udp -m state --state NEW -m multiport --dports
875,890,892,12025 -j ACCEPT
# iptables -I INPUT -p tcp -m state --state NEW -m multiport -dports
875,890,892,12025 -j ACCEPT
# service iptables save
```

Now NFSv3 needs to be reconfigured to statically assign ports for its different components, so that the previous firewall changes can be effective. Confirm the changes by making sure the output lines of the grep command are not commented out and the port numbers specified match those in the previous iptables commands. On each of the servers, apply the following changes:

```
# sed -i 's/.*RQUOTAD_PORT=.*/RQUOTAD_PORT=875/' /etc/sysconfig/nfs
# sed -i 's/.*LOCKD_UDPPORT=.*/LOCKD_UDPPORT=890/' /etc/sysconfig/nfs
# sed -i 's/.*LOCKD_TCPPORT=.*/LOCKD_TCPPORT=890/' /etc/sysconfig/nfs
# sed -i 's/.*MOUNTD_PORT=.*/MOUNTD_PORT=892/' /etc/sysconfig/nfs
# sed -i 's/.*STATD_PORT=.*/STATD_PORT=12025/' /etc/sysconfig/nfs
# egrep "D_PORT|PPORT" nfs
RQUOTAD_PORT=875
```

```
LOCKD_TCPPORT=890
LOCKD_UDPPORT=890
MOUNTD_PORT=892
STATD_PORT=12025
```

At this point the servers are ready to accept NFSv3 traffic through the firewall.

Alternately, turn off the firewall. Ensure that both public and private interfaces are on a secure network and be aware of the security implications of turning off the firewall before implementing this alternative:

```
# service iptables stop; chkconfig iptables off
# service ip6tables stop; chkconfig ip6tables off
```

A.3.6. Configure startup services.

```
# chkconfig ipmi on
# service ipmi start
# chkconfig nfs on
```

In case of failure, the following three rpms need to be installed

```
OpenIPMI-2.0.16-12.el6.x86_64.rpm
nfs-utils-1.2.3-7.el6.x86_64.rpm
```

A.3.7. Configure fence devices

1. Configure iDRAC on each R710, and make sure the login name, password, and IP address are configured according to the NSS-HA cluster specification in section A.1.1.
    a) Make sure iDRAC is installed.
    b) Boot R710. On boot, when prompted hold **<CTRL-E>**
    c) Verify the parameter **idrac LAN = on**
    d) Navigate to **LAN Parameters** and press **ENTER**
    e) Scroll down to **IPv4 Settings** and set the IP Address

    ```
    IPV4 Address            15.15.10.20x
    Subnet Mask             255.255.255.0
    Default Gateway         xxx.xxx.xxx.xxx
    ```

    f) Press **ESC** once to return to the main menu.
    g) Scroll down to **LAN USER CONFIGURATION** and press **ENTER**.
    h) Navigate to **the Account User Name** section and modify the parameters:

    ```
    Account User Name [root          ]
    Account Password  [              ]
    Verify Password   [              ]
    ```

    i) Press **ESC** twice to exit and continue booting.

2. Configure two APC PDUs, and make sure the login name, password, and IP address are configured according to the NSS-HA cluster specification in section A.1.1.

    a) Connect with a serial cable and run a terminal program such as `putty` or `minicom` using settings 9600/8/N/N

    b) Press **ENTER** and login with the default username and password

        Default user name: apc
        Default password: apc

    c) Configure the network:

        From the **Control Console** select options:

        2- Network
        1- TCP/IP

        From here, set the System IP, Subnet Mask, and Gateway:

        1- System IP : 15.15.10.10x
        2- Subnet Mask : 255.255.255.0
        3- Default Gateway: xxx.xxx.xxx.xxx
        4- Boot Mode : Manual

    d) Press **<ESC>** twice to return to the Control Console

    e) Changing the default username

        From the **Control Console**, select:

        3- System
        1- User Manager
        1- Administrator

        Enter the current Administrator password when prompted. From here you can customize the default user name and password (apc/apc).

    f) When finished, select option:

        5- Accept Changes :

    g) Press **<ESC>** twice to return to the Control Console menu.

Note: Please double check iDRAC and APC PDU manuals.

A.3.8. Checklist

Before moving to the next section, please make sure all following tasks are completed.

Tasks	Notes
✓ Install OS, configure swap disks, and install XFS packages	Section A.3.1
✓ Configure multipath	Section A.3.2. Please make sure root device is not in the control of multipath, it is very important for the future HA cluster configuration.
✓ Install Mellanox package and set network IPs.	Section A.3.3. If 10gbE is deployed, please skip the part of installing Mellanox package.
✓ Install OSMA and storage management tool	Section A.3.4. Please make sure that the storage management tool is installed after the configuration of multipath, it is very important.
✓ Network security setting	Section A.3.5.
✓ Configure startup service	Section A.3.6.
✓ Configure and test the configuration of PDUs and iDRAC.	Section A.3.7. Please make sure there is no problem to remote login PDUs via Telnet, and no problem to remote login iDRAC via ssh.
✓ Check the software versions, firmware versions, and driver versions.	Refer to section 5.2, Table 6 and 7

A.4. Performance tuning on the server

All the operations below apply to each R710.

1. If the clients access the NFS server via 10GbE, configure the MTU on the 10GbE device to be 8192 for both the active and the passive server. Note that the switches need to be configured to support large MTU as well.

   On the server, if the value of MTU is not specified in /etc/sysconfig/network-scripts/ifcfg-p2p1,

   ```
   # echo "MTU=8192" >> /etc/sysconfig/network-scripts/ifcfg-p2p1
   ```

   change the old value to 8192 in `/etc/sysconfig/network-scripts/ifcfg-p2p1.`

   where ifcfg-p2p1 is the 10GbE network interface.

   Restart the networking services.

   ```
   # service network restart
   ```

2. On both the active and the passive server, change the number of NFS threads from a default of 8 to 256.

   Make a backup of /etc/sysconfig/nfs and change the number of threads

   ```
   # cp /etc/sysconfig/nfs{,.orig}
   # sed -i 's/#RPCNFSDCOUNT=8/RPCNFSDCOUNT=256/' /etc/sysconfig/nfs
   ```
   Restart the NFS service.

   ```
   # service nfs restart
   ```

   Reference - DELL™ PowerVault™ MD1200 Performance as a Network File System (NFS) Backend Storage Solution.
   http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/hpc-pv-md1200-nfs.pdf

3. On both the active and the passive server, change the OS I/O scheduler to "`deadline`".

   To the end of the kernel line in `/etc/grub.conf`, add `elevator=deadline`

   On both the active and the passive servers, preload the driver to avoid excessive messages at boot time. Find the kernel line in /etc/grub.conf, and append `rdloaddriver=scsi_dh_rdac`

   A reboot the server is needed for this change to take effect, however DO NOT REBOOT until section A6 is completed.

4. Due to the performance issues observed with NFSv4, some users may want to use NFSv3 to access the NSS. Two options are proposed to accomplish such restriction.

   One option is to preclude NFS version 4 on each of the NSS servers:

   Edit the file `/etc/sysconfig/nfs` and uncomment the `RPCNFSDARGS` line below

   ```
   # Turn off v4 protocol support
   RPCNFSDARGS="-N 4"
   ```

Restart the NFS service, `service nfs restart`

Run the `nfsstat` command. It should show output only for v3 and not for v4.

Another option is make the change on the clients. On each of the clients, mount the NFS share using the option "`-o vers=3`". This will use NFSv3 for the clients. If the server supports NFSv4, the default mount option without an explicit `vers=3` parameter will be NFSv4.

A.4.1. Checklist

Before moving to the next section, please make sure all following tasks are completed.

Tasks	Notes
✓ Set MTU for 10gbE link	Section A.4, step 1. If IB network is deployed, please skip this.
✓ Configure NFS	Section A.4, step 2 and step 4.
✓ Configure server I/O scheduler.	Section A.4, step 3.

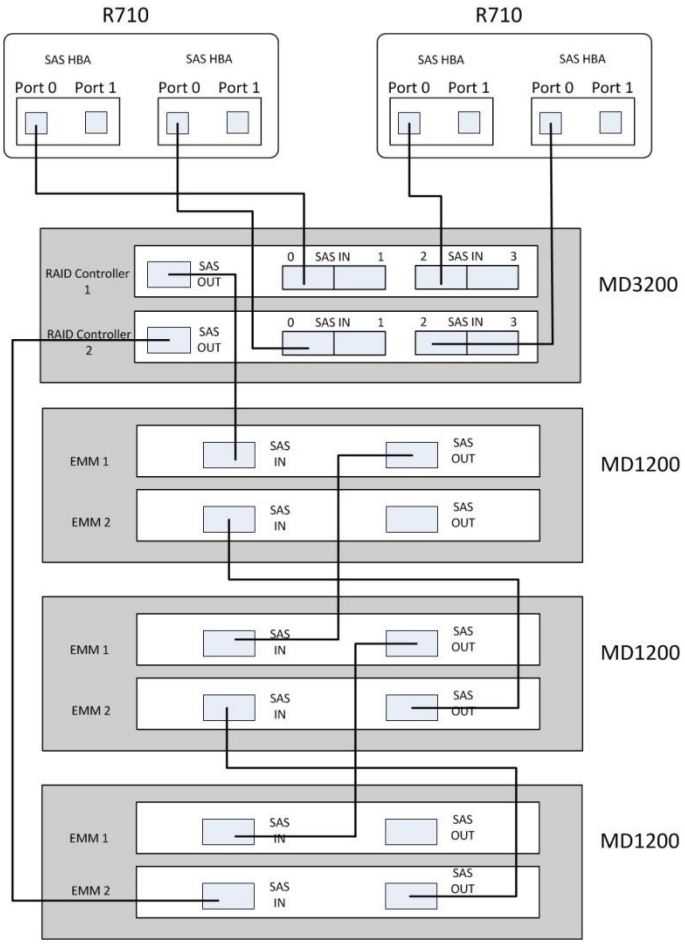
A.5. Storage hardware setup

1. Cable the MD3200 to the SAS 6 Gbps card on servers as shown in the figure below. Each server has two dual-port SAS cards. Cable one port on each SAS card to the storage. That is, each server will have one cable per SAS card going to the MD3200.

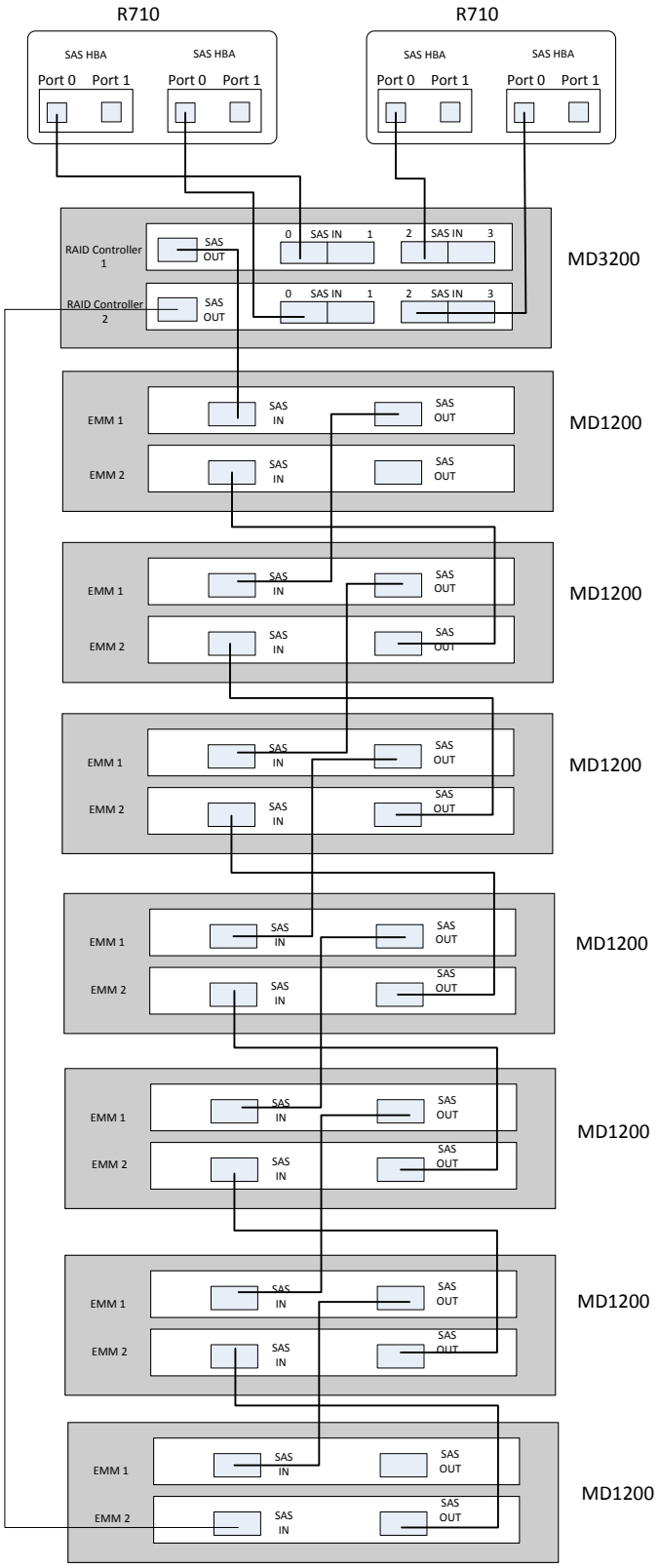
   Reference: Dell PowerVault MD3200 and MD3220 Storage Arrays Deployment Guide, http://support.dell.com/support/edocs/systems/md3200/en/DG/PDF/DG.pdf, Section "Dual Controller Configurations"

   Ensure that each server is connected to both the MD3200 RAID controllers as shown in the figure below.

   - Cabling for 144TB configuration



- Cabling for 288TB configuration



Note: Do not cable storage to R710 before OS install, and use the leftmost port of each SAS card on the two R710s.

2. To cable MD1200s to MD3200, please refer to the figure above.

   Reference: Quick SAS Cabling Guide --A Dell Technical White Paper.
   http://www.dell.com/downloads/global/products/pvaul/en/powervault-md3200-m3200i-cabling-guide.pdf

3. Power on all MD3200 and MD1200s.

A.6. Storage configuration

1. Launch the MDSM management GUI on one R710. Discover the attached storage array via in-band management and add the storage array to the management GUI.
2. Create a host group (named as `NSS-HA61)` and add the active and passive servers to the group.
3. Create a disk group (named as `NSS_HA_MD3200`) and virtual disks on storage using MDSM GUI
   - For each array, create a disk group and virtual disks that contain all 12 disks with a configuration of a 10+2 RAID6 with a segment size of 512k, respectively. For the NSS-HA 144TB configuration, this is a total of four virtual disks in a disk group. For NSS-HA 288TB configuration, this is a total of eight virtual disks in a disk group. Note that virtual disk initialization can take between 24 to 30 hours.
   - Enable read cache, write cache, write cache mirroring, and dynamic cache read prefetch for each virtual disk.
   - Enable the high performance tier on the MD3200.
   - Set the cache block size to 32k.

   For instruction on how to configure the storage array, see the Dell PowerVault MD3200 and MD3220 Storage Arrays Owner's Manual:
   http://support.dell.com/support/edocs/systems/md3200/en/OM/PDF/MD3200.pdf

   and please also refer to Dell™ PowerVault™ Modular Disk Storage Manager User's Guide
   http://support.dell.com/support/edocs/systems/md3000/en/1stgen/ug/pdf/uga00mr.pdf

4. Map all the virtual disks to the host group `NSS-HA61` that contains the active and passive servers.

   Reference: Dell PowerVault MD3200 and MD3220 Storage Arrays Owner's Manual,
   http://support.dell.com/support/edocs/systems/md3200/en/OM/PDF/MD3200.pdf

5. Once the disk group and virtual disks are successfully created and initialized, **REBOOT the two R710s**. Once the two servers are back up, and then execute the following command on each of them.

   ```
   # SMcli -n NSS_HA_MD3200 -c "show storagearray profile;"
   ```

   Check the information in the `Host Group` section.

```
Host Group:                    NSS-HA61
           Host:                          active
               Host type:                 Linux
               Interface type:            SAS
                  Host port identifier:   5a:4b:ad:b0:47:ba:98:00
                  Alias:                  activeP0
                  Host port identifier:   5a:4b:ad:b0:50:d6:4d:00
                  Alias:                  activeP1
            Host:                         passive
               Host type:                 Linux
               Interface type:            SAS
                  Host port identifier:   57:82:bc:b0:39:fd:62:00
                  Alias:                  passiveP0
                  Host port identifier:   57:82:bc:b0:39:fd:54:00
                  Alias:                  passiveP1
```

Make sure the host port identifiers on each R710 match the `sas_address` from `/sys/class/sas_phy/phy-1:0/sas_address` and `/sys/class/sas_phy/phy-2:0/sas_address`.

6. On each R710, run the command `rescan_dm_devs` to detect all the virtual disks.
7. On each R710, `cat /proc/partitions` and `multipath -ll` should show all the LUNs on the storage.

   Reference: Configuration: Device Mapper Multipath for Linux, http://support.dell.com/support/edocs/systems/md3200/en/OM/HTML/config_n.htm
8. Make sure each vitual disk on two R710s has the same wwid. For example, check the output of `multipath -ll | grep MD32xx`, and every device with same name should have same wwids, as below:

```
mpatha (36842b2b0004c256400000db24e95a162) dm-4 DELL,MD32xx (from active)
mpatha (36842b2b0004c256400000db24e95a162) dm-4 DELL,MD32xx (from passive)
```

   If the wwids do not match, execute the following commands one by one:

   a. On machine named as “passive”:

```
# service multipathd stop
# multipath -F
```

   b. Copy /etc/bindings from machine named as “active” to “passive”

   c. On “passive”:

```
# service multipathd start
```

NOTE:
Under certain conditions, on solutions with many arrays (288 TB and bigger), `multipathd` may start before all the virtual disks from the MD3200 are detected and ready to be used, resulting in missing links under `/dev/mapper` and other problems related to those missing links.

A way to check for this problem is to compare the number of virtual disks or LUNs your NSS has with the number of `/dev/mapper` links. As an example, a 288 TB solution must have 8 links:

```
# ls -l /dev/mapper | grep mpath | wc -l
8
```

If this problem occurs (number of links < 8), a workaround is to send the service a reload after a couple minutes, to make sure all the virtual disks are already visible to multipathd.
Edit the file `/etc/rc.local` on both servers and append the line:

```
sleep 120; service multipathd reload
```

Please note that this workaround will slow down the server boot time by 120 seconds

A.6.1. Checklist

Before moving to the next section, please make sure all following tasks are completed.

Tasks		Notes
✓	Configure the storage array	Section A.6, step 1, 2, 3, and 4.
✓	Check the storage array is correctly configured	Section A.6, step 5, 6, 7 and 8. It is very important to make sure all the four steps are successfully completed.

A.7. NSS HA cluster setup

In this recipe the term “cluster” refers to the active-passive NSS-HA Red Hat cluster.

A.7.1. Prepare

1. On both R710s install the cluster software packages.

```
#  yum install -y ricci rgmanager cman openais lvm2-cluster ccs
#  service ricci start; chkconfig ricci on
```

2. Set a password for user ricci using the command below

```
# passwd ricci
```

3. Create a mount point for the file system.

   On both servers, create a mount point for the XFS file system. This is the directory where the XFS file system will be mounted and that will be exported over NFS to the clients.

```
# mkdir /mnt/xfs1
```

   If you plan to use the storage for home directories, you also need to configure SElinux to allow it.

```
# chcon -t home_root_t /mnt/xfs1
```

   Similarly, the clients need SElinux to allow NFS home directories.

```
# setsebool -P use_nfs_home_dirs 1
```

   Alternatively, you can use the following command to disable SElinux on the clients, the server or both. A reboot is needed for the change to take effect.

```
# sed -i 's/.*SELINUX=.*/SELINUX=disabled/' /etc/sysconfig/selinux
```

4. For InfiniBand clusters, copy the `ibstat_script.sh` file provided in Section A.11 to `/root/ibstat_script.sh` on both servers. This script checks the InfiniBand link status using the `ibstat` command. It is included as a resource of the cluster service to ensure that InfiniBand link is monitored. (For 10 Gigabit Ethernet clusters, RHCS monitors the 10GbE link and **no additional scripts are needed**).

5. Copy the `sas_path_check.sh` file provided in Section A.11 to `/root/sas_path_check.sh` on both servers as shown below. This script checks the status of a device on the shared storage. If the device is not accessible within a set period of time, the server will reboot prompting a failover of the cluster service to the other server. The reboot will trigger if the cluster service is unable to stop gracefully because of a failed SAS path.

   The device to check (i.e. `/dev/mapper/mpatha`) and the timeout period (300 second) are tunable. Make sure that the device in the script, e.g., `/dev/mapper/mpatha` points to a LUN on the shared MD3200 storage, **otherwise, please modify the device name**. This can be checked by looking at the output of the `multipath -ll` command.

```
# cp sas_path_check.sh /root/sas_path_check_1.sh
```

6. Check that the public interface is up on both servers. This is the 10GbE link or the InfiniBand link.

   For 10GbE, using

   ```
   # ethtool p2p1 | grep "Link detected"
   ```

   If the link is up, the output will display "`Link detected: yes`".

   For InfiniBand, using

   ```
   # ibstat | grep "Physical state"
   ```

   If the link is up, the output will display "`Physical state: LinkUp`"

7. Check /etc/lvm/lvm.conf, make sure `locking_type` is set to `3`.

   ```
   # Type 3 uses built-in clustered locking.
   # Type 4 uses read-only locking which forbids any operations that might
   # change metadata.
   locking_type = 3
   ```

8. Check the two fence devices: iDrac and APC PDU.

   For iDrac, using

   ```
   # fence_drac5 -o status -a 15.15.10.201 -l root -p calvin -d 5 -x -c admin1
   # fence_drac5 -o status -a 15.15.10.202 -l root -p calvin -d 5 -x -c admin1
   ```

   Expect to see the output of `STATUS: ON`.

   For APC PDU, using

   ```
   # fence_apc -o status -a 15.15.10.101 -l apc -p apc -n 2
   # fence_apc -o status -a 15.15.10.101 -l apc -p apc -n 3
   # fence_apc -o status -a 15.15.10.102 -l apc -p apc -n 2
   # fence_apc -o status -a 15.15.10.102 -l apc -p apc -n 3
   ```

   Expect to see the output of `STATUS: ON`.

A.7.2. Create HA cluster

Before the actual HA cluster construction, please make sure that all steps in section A.7.1 are successfully completed, otherwise, unexpected issues will appear during the HA cluster construction.

There are two ways to create and configure Red Hat HA cluster: with Conga GUI and with ccs commands. For the sake of simplicity, the steps for configuring HA cluster via ccs commands are provided via a script (`/root/config_cluster.sh`).

Refer to Red Hat Enterprise Linux 6 Cluster Administration -- Configuring and Managing the High Availability Add-On. http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Cluster_Administration/Red_Hat_Enterprise_Linux-6-Cluster_Administration-en-US.pdf, section 4 and 5.

And, a very good FAQ on Red Hat Cluster Suite is at http://sources.redhat.com/cluster/wiki/FAQ .

1. Generate cluster configuration file.

   On the 'active' R710, manually modify the cluster configuration script according to the cluster spec.

   In /root/config_cluster.sh

   Modify:

```
machine_active="active"                 #check section A.1.1
machine_passive="passive"               #check section A.1.1

pdu1_ip="15.15.10.101"                  #check section A.1.1
pdu2_ip="15.15.10.102"                  #check section A.1.1
pdu_active_port="2"                     #check section A.1.1
pdu_passive_port="3"                    #check section A.1.1

drac_active_ip="15.15.10.201"           #check section A.1.1
drac_passive_ip="15.15.10.202"          #check section A.1.1


cluster_name="NSS61"    #define the name of your cluster
num_HA=1                #define the number of HA services
set_IB=0                #if IB is used, set to 1.

HA_ip_1=10.10.10.200                    #check section A.1.1
```

   Then, execute the script

```
# /root/config_cluster.sh
```

   Very important: once the cluster configuration file is generated, please double check the configuration file to make sure all information is correct according to the specification in section A.1.1. The generated configuration file is /etc/cluster/cluster.conf.

2. On both servers, start and check that the cman service is running:

```
# service cman start
# service cman status
```

3. On both servers, check the cluster status:

```
# clustat
```

   Cluster status should show both servers online. If the cluster is up and running, then go on to the next step.

```
[root@active ~]# clustat

Cluster Status for NSS61 @ Wed Jan  4 17:36:36 2012
Member Status: Quorate
Member Name                ID   Status
```

```
------ ----                 ---- ------
active                      1    Online, Local
passive                     2    Online
```

4. Start `clvmd` service on both servers to prepare the creation of physical volumes, volume group and logical volume in an HA cluster. Make sure `locking_type` is set to 3 in `/etc/lvm/lvm.conf` before executing the following command.

   ```
   # service clvmd start
   ```

   And make sure that service `multipathd` is running.

5. Once `clvmd` is started, please execute the script `nssha61_single.py` to automatically configure the shared storage array including creating physical volumes, volume group, logic volume, and XFS file system.

   ```
   # /root/nssha61_single.py
   ```

   Note: section A.9.5 provides instructions to manually configure the shared storage array.

   Use the following commands to verify that PVs, VG, LV, and XFS file system are successfully created.

   ```
   # df -h
   # lvs
   # vgs
   # pvs
   ```

   The names and size for PVs, VG, LV, XFS file system should be observed from the outputs of the above commands.

   Then, unmount XFS on `/mnt/xfs1`.

   ```
   # umount /mnt/xfs1
   ```

6. Before starting the HA service, logical volume should be deactivated on the active server.

   ```
   # lvchange -an /dev/VGMD1/LVMD1
   ```

7. Now start HA service on both servers.

   ```
   # service modclusterd start
   # service rgmanager start
   ```

A.7.3. Verification

1. Check that the cluster service is running.

   Cluster status should show both servers online. If the cluster is up and running, then go on to the next step. `HA1` is the name of the defined HA service.

```
[root@active ~]# clustat
Cluster Status for NSS61 @ Thu Jan  5 10:41:48 2011
Member Status: Quorate
Member Name                              ID   Status
------ ----                              ---- ------
active                                   1    Online, rgmanager
passive                                  2    Online, Local, rgmanager

Service Name                    Owner (Last)                    State
------- ----                    ----- ------                     -----
service:HA1                     active                          started
```

On the server that is running the service, check that the XFS file system is mounted.

```
[root@active ~]# mount | grep xfs1

/dev/mapper/VGMD1-VGMD1-LVMD1 on /mnt/xfs1 type xfs
(rw,noatime,allocsize=1g,nobarrier,inode64,logbsize=262144,wsync)
```

On the server that is running the service, check that the resource IP is assigned. The interface to the public network should have two IP addresses - the statically assigned address and the floating service IP address. For example,

```
[root@active ~]# ip addr show ib0
9: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc pfifo_fast qlen
256
link/infiniband
80:00:00:48:fe:80:00:00:00:00:00:00:02:c9:03:00:07:7f:a7 brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
inet 10.10.10.201/24 brd 10.10.10.255 scope global ib0
inet 10.10.10.200/24 scope global secondary ib0
inet6 fe80::202:c903:7:7fa7/64 scope link
valid_lft forever preferred_lft forever
```

In order to stop the cluster, execute the following commands on both servers:

```
# clusvcadm –d HA1
# service clvmd stop
# service rgmanager stop
# fence_tool leave
# cman_tool leave remove
# service modclusterd stop
# service cman stop
```

2. Update the Selinux policy. This is needed to allow a cluster server to fence the other cluster member and take ownership of the cluster service.

An SELinux policy can be generated from logs of denied operations. Check `/var/log/audit/audit.log` for denied operations. If there are none relating to the cluster, test fencing as described in the "Quick test of HA set-up" section and then follow the steps below.

```
# grep avc /var/log/audit/audit.log | audit2allow -M NSSHApolicy
```

Install the module on bot servers

```
# semodule -i NSSHApolicy.pp
```

Reference https://bugzilla.redhat.com/show_bug.cgi?id=588902

3. On both servers configure the cluster startup service

```
# chkconfig cman on
# chkconfig clvmd on
# chkconfig rgmanager on
# chkconfig modclusterd on
```

A.8. Quick test of HA setup

1. Test fencing:
   - First disable the cluster service:

     ```
     [root@active ~]# clusvcadm –d HA1
     ```

   - From the active server run the command `fence_node passive`. This should power cycle the passive server via DRAC. Check /var/log/messages on active.
   - From passive run the command `fence_node active`. This should power cycle the active server via iDRAC. Check `/var/log/messages` on passive.
   - Disconnect the DRAC cable on passive. From “active” run the command `fence_node passive`. This should power cycle the passive server via the switched PDU.
   - Disconnect the DRAC cable on active. From passive run the command `fence_node active`. This should power cycle the “active” server via the switched PDU.
2. Simple HA failover test:
   - Start the service on the “active” server:

     ```
     [root@active ~]# clusvcadm –e HA1 –m active
     ```

   - Power down the active server by holding down the power button or by pulling out both power cords.
   - Service should migrate to passive. Watch `/var/log/messages` and check `clustat` status on passive.

A.9. Useful commands and references

This section provides several commands for HA cluster configuration, management, and debug, and also gives the instructions to configure a storage array manually.

A.9.1.Manually modify cluster configuration file

If the `/etc/cluster/cluster.conf` file is edited manually, make the changes only on one server and increment the version number field. Use `cman_tool` command to synchronize the `/etc/cluster/cluster.conf` file on both servers.

For example, if there is a requirement to change the service IP for the public network (floating IP that the clients will use to mount NFS), please follow the instructions below. (Assume that the old IP is 10.10.10.200, and the new IP is 10.10.10.100).

a. On one server, edit `/etc/cluster/cluster.conf`

b. Locate the old IP in the file, it appears in two places.

```
<ip address="10.10.10.200" monitor_link="on" sleeptime="10"/>
<ip ref="10.10.10.200"/>
```

c. Make the change.

```
<ip address="10.10.10.100" monitor_link="on" sleeptime="10"/>
<ip ref="10.10.10.100"/>
```

d. Increase the `config_version` number (gotten from the current `/etc/cluster/cluster.conf`) by one. For example, if the current `config_version` is 16, change it to 17.

```
<cluster config_version="17" name="NSS61">
```

e. Validate the edited configuration file.

```
# ccs_config_validate
```

f. If there is no error, on the server, run the following command to synchronize the cluster configuration file between the two servers.

```
# cman_tool version -r
```

A.9.2.Manually stop, disable, start and relocate the cluster service

1. Stop cluster service. Assume the service name is `HA1`.

```
# clusvcadm -s HA1
```

2. Disable cluster service. Assume the service name is `HA1`.

```
# clusvcadm -d HA1
```

3. Start cluster service. Assume the service name is `HA1`, it is disabled, and will be started on "passive" server.

```
# clusvcadm –e HA1 –m passive
```

4. Relocate cluster service. Assume the service name is `HA1`, it is running on "passive" server, and will be relocated to "active" server.

```
# clusvcadm –r HA1 –m active
```

A.9.3. Debug HA cluster configuration

If a cluster-controlled service will not start, it may be due to configuration or syntax faults in the service configuration as specified `in /etc/cluster/cluster.conf`. `rg_test` is a powerful debug tool to identify the sources of issues.

Note: Before running `rg_test`, make sure that the cluster service is stopped and disabled first.

For example, assume the cluster service name is `HA1`, to debug the HA cluster configuration, please follow the instructions below.

```
# clusvcadm –s HA1
# clusvcadm –d HA1
# rg_test test /etc/cluster/cluster.conf
```

Please check the output of `rg_test` to verify a configuration or identify issues with a configuration.

`rg_test` can also be used to explicitly start or stop a service to display the start or stop ordering of a service. For example,

Start a service

```
# rg_test test /etc/cluster/cluster.conf start service HA1
```

Stop a service

```
# rg_test test /etc/cluster/cluster.conf stop service HA1
```

Note: Only do the above tests on one node, and always disable the service using clusvcadm first. Clients will be able to access the file system when `rg_test` is used to start a service since `rg_test` mimics normal operation. So use `rg_test` with care and ONLY for debugging. `clustat` and `clusvcadm` should be used to manage the cluster and cluster service during normal operation.

A.9.4. Remove a node from HA cluster

To turn off or reboot the "active" or "passive" server, it must first leave the HA cluster gracefully. Otherwise the other server will assume that it has died and fence it. To leave the cluster gracefully, follow the instructions below.

```
# service clvmd stop
# service rgmanager stop
# fence_tool leave
# cman_tool leave remove
```

```
# service cman stop
```

A.9.5.Configure the shared storage array manually

If manual configuration of the storage array configuration is preferred instead of the using the script "nssha61_single.py" mentioned in Section A.7.2, step 5 , please follow the instructions below.

1. Before configuring a shared storage array, please make sure service `clvmd` is running on both servers, and make sure the step 7 in section A.7.1 is completed.
2. On one of the servers (in this example, "active"), create the physical volumes, volume group, logical volume, and the file system.

Configure a 144TB storage array:

a. Obtain `mpath` names from the output of `multipath -ll`. All `mpath` names should exist in `/dev/mapper/`.

b. Create physical volumes.

```
# pvcreate /dev/mapper/mpatha /dev/mapper/mpathb /dev/mapper/mpathc
/dev/mapper/mpathd
```

Note: different mpath names may be observed according to the actual configuration, please modify them before executing `pvcreate`.

c. Create volume group.

For creating volume group, please pay attention to the order of physical volumes, making sure they follow the order of LUN ids (from small to large). For example, if `multipath -ll` shows that

```
mpathd (36842b2b0004c256400000db74e95a1cb) dm-7 DELL,MD32xx
size=27T features='3 queue_if_no_path pg_init_retries 50' hwhandler='1
rdac' wp=rw
|-+- policy='round-robin 0' prio=6 status=active
| `- 2:0:0:2 sdm 8:192 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 1:0:0:2 sde 8:64  active ghost running
mpathc (36842b2b0004c21e00000072b4e95a6b0) dm-1 DELL,MD32xx
size=27T features='3 queue_if_no_path pg_init_retries 50' hwhandler='1
rdac' wp=rw
|-+- policy='round-robin 0' prio=6 status=active
| `- 1:0:0:3 sdf 8:80  active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 2:0:0:3 sdn 8:208 active ghost running
mpathb (36842b2b0004c21e0000007284e95a64a) dm-3 DELL,MD32xx
size=27T features='3 queue_if_no_path pg_init_retries 50' hwhandler='1
rdac' wp=rw
|-+- policy='round-robin 0' prio=6 status=active
| `- 1:0:0:1 sdd 8:48  active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 2:0:0:1 sdl 8:176 active ghost running
mpatha (36842b2b0004c256400000db24e95a162) dm-4 DELL,MD32xx
```

```
size=27T features='3 queue_if_no_path pg_init_retries 50' hwhandler='1
rdac' wp=rw
|-+- policy='round-robin 0' prio=6 status=active
| `- 2:0:0:0 sdk 8:160 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 1:0:0:0 sdc 8:32  active ghost running
```

The LUN id is the last number of `2:0:0:x` or `1:0:0:x`.

The order should be `mpatha, mpathb, mpathd, mpathc.`

```
# vgcreate VGMD1 /dev/mapper/mpatha /dev/mapper/mpathb /dev/mapper/mpathd
/dev/mapper/mpathc
```

d. Create logic volume.

```
# lvcreate -i 4 -I 1024 -l 100%FREE VGMD1  -n LVMD1
```

e. Create XFS file system.

```
# mkfs.xfs -l size=128m /dev/VGMD1/LVMD1
```

f. Mount XFS file system.

```
# mount -o noatime,allocsize=1g,nobarrier,inode64,logbsize=262144,wsync
/dev/VGMD1/LVMD1 /mnt/xfs1/
```

Make sure the above commands can be executed successfully, then unmount XFS on `/mnt/xfs1.`

g. Unmount XFS file system.

```
# umount /mnt/xfs1
```

h. Go to step 6 in section A.7.2 to resume the HA cluster configuration.

Configure 288TB storage array

For 288TB storage array configuration, it is very similar to the 144TB storage array configuration. The difference is that 288TB configuration requires performing 144TB configuration first, then extending the configuration to 288TB.

```
# pvcreate /dev/mapper/mpatha /dev/mapper/mpathb /dev/mapper/mpathc
/dev/mapper/mpathd /dev/mapper/mpathe /dev/mapper/mpathf
/dev/mapper/mpathg /dev/mapper/mpathh
```

Note: different mpath names may be observed according to the actual configuration, please modify them before executing `pvcreate`.

For creating volume group, pay attention to the order of physical volumes, make sure they follow the order of LUN ids (from small to large).

a. Configure 144TB storage array.

```
# vgcreate VGMD1 /dev/mapper/mpatha /dev/mapper/mpathb /dev/mapper/mpathd
/dev/mapper/mpathc
# lvcreate -i 4 -I 1024 -l 100%FREE VGMD1  -n LVMD1
# mkfs.xfs -l size=128m /dev/VGMD1/LVMD1
# mount -o noatime,allocsize=1g,nobarrier,inode64,logbsize=262144,wsync
/dev/VGMD1/LVMD1/mnt/xfs1/
```

b. Extend 144TB configuration to 288TB.

```
# vgextend VGMD1 /dev/mapper/mpathe /dev/mapper/mpathf /dev/mapper/mpathg
/dev/mapper/mpathh

# lvextend /dev/VGMD1/LVMD1 /dev/mapper/mpathe /dev/mapper/mpathf
/dev/mapper/mpathg /dev/mapper/mpathh –I 1024 –i 4

# xfs_growfs -d /mnt/xfs1
```

c. Un-mount XFS file system.

```
# umount /mnt/xfs1
```

d. Go to step 6 in section A.7.2 to resume the HA cluster configuration.

A.10. Performance tuning on clients

1. If the clients access the NFS server via 10GbE, configure the MTU on the 10GbE device to be 8192 for all the clients. Note that the switches need to be configured to support large MTU as well.

   On the client, if the value of MTU is not specified in `/etc/sysconfig/network-scripts/ifcfg-p2p1`,

   ```
   # echo "MTU=8192" >> /etc/sysconfig/network-scripts/ifcfg-p2p1,
   ```

   Otherwise change the old value to 8192 in `/etc/sysconfig/network-scripts/ifcfg-p2p1`.

   where `ifcfg-p2p1` if the 10GbE network interface.

   Restart the networking services.

   ```
   # service network restart
   ```

2. If the clients access the NFS server via 10GbE, the Ethernet switches on the fabric should have flow control disabled. The following instructions are for the PowerConnect 6248 and the PowerConnect 8024 and can be used as a reference.

   Start a console session to the switch and type the following commands in sequence:

   ```
   console>enable
   console#configure
   console(config)#no flowcontrol
   ```

   In order to verify the setting of the option, type the following command:

   ```
   console#show interface status
   ```

   A message of "`Flow Control:Disabled`" will be printed out.

3. For each client, Add the following to `/etc/sysctl.conf`

   Increasing the default TCP receive memory size

   ```
   net.ipv4.tcp_rmem = 4096 2621440 16777216
   ```

   Activate the changes with `sysctl -p`.

A.11. Scripts

1. `/root/config_cluster.sh` file for automatically generating cluster configuration file
2. `/root/nssha61_single.py` file for automatically configuring the shared storage.
3. `/root/ibstat_script.sh` file for InfiniBand clusters
4. `/root/sas_path_check.sh` file

Appendix B: Benchmarks and test tools

The `iozone` benchmark was used to measure sequential read and write throughput (MB/sec) as well as random read and write I/O operations per second (IOPS).

The `mdtest` benchmark was used to test metadata operation performance.

The `checkstream` utility was used to test for data correctness under failure and failover cases.

The Linux `dd` utility was used for initial failover testing and to measure data throughput as well as time to complete for file copy operations.

B.1. IOzone

`IOzone` can be downloaded from http://www.iozone.org/. Version 3.397 was used for these tests and installed on both the NFS servers and all the compute nodes.

The `IOzone` tests were run from 1-64 nodes in clustered mode. All tests were N-to-N, i.e. N clients would read or write N independent files.

Between tests, the following procedure was followed to minimize cache effects:

- Unmount NFS share on clients.
- Stop the cluster service on the server. This umounts the XFS file system on the server.
- Start the cluster service on the server.
- Mount NFS Share on clients.

The following table describes the IOZone command line arguments.

IOzone ARGUMENT	DESCRIPTION
-i 0	Write test
-i 1	Read test
-i 2	Random Access test
-+n	No retest
-c	Includes close in the timing calculations
-t	Number of Threads
-e	Includes flush in the timing calculations
-r	Records size
-s	File size
-t	Number of Threads
+m	Location of clients to run IOzone on when in clustered mode
-w	Does not unlink (delete) temporary file
-I	Use O_DIRECT, bypass client cache
-O	Give results in ops/sec.

For the sequential tests, file size was varied along with the number of clients such that the total amount of data written was 128G (number of clients * file size per client = 128G).

IOzone Sequential Writes

```
# /usr/sbin/iozone -i 0 -c -e -w -r 1024k -s 2g -t 64 -+n -+m ./clientlist
```

IOzone Sequential Reads

```
# /usr/sbin/iozone -i 1 -c -e -w -r 1024k -s 2g -t 64 -+n -+m ./clientlist
```

For the random tests, each client read or wrote a 2G file. The record size used for the random tests was 4k to simulate small random data accesses.

`IOzone` IOPs Random Access (Reads and Writes)

```
# /usr/sbin/iozone -i 2 -w -r 4k -I -O -w -+n -s 2G -t 1 -+m ./clientlist
```

By using `-c` and `-e` in the test, `IOzone` provides a more realistic view of what a typical application is doing. The `O_Direct` command line parameter allows us to bypass the cache on the compute node on which we are running the `IOzone` thread.

B.2. mdtest

`mdtest` can be downloaded from http://sourceforge.net/projects/mdtest/. Version 1.8.3 was used in these tests. It was compiled and installed on a NFS share that was accessible by compute nodes. `dtest` is launched with `mpirun`. For these tests, MPICH2 version 1.3.2 was used. The following table describes the `mdtest` command line arguments.

mpirun ARGUMENT	DESCRIPTION
-np	Number of Processes
--nolocal	Instructs mpirun not to run locally
--hostfile	Tells mpirun where the hostfile is
mdtest ARGUMENT	**DESCRIPTION**
-d	The directory mdtest should run in
-i	The number of iterations the test will run
-b	Branching factor of directory structure
-z	Depth of the directory structure
-L	Files only at leaf level of tree
-I	Number of files per directory tree
-y	Sync the file after writing
-u	unique working directory for each task
-C	Create files and directories
-R	Randomly stat files
-T	Only stat files and directories
-r	Remove files and directories left over from run

As with the IOzone random access patterns, the following procedure was followed to minimize cache effects during the metadata testing:

- Unmount NFS share on clients.
- Stop the cluster service on the server. This umounts the XFS file system on the server.
- Start the cluster service on the server.
- Mount NFS Share on clients.

Metadata file and directory creation test:

```
# mpirun -np 32 --nolocal --hostfile ./hosts /nfs/share/mdtest -d
/nfs/share/filedir -i 6 -b 320 -z 1 -L -I 3000 -y -u -t -C
```

Metadata file and directory stat test:

```
# mpirun -np 32 --nolocal --hostfile ./hosts /nfs/share/mdtest -d
/nfs/share/filedir -i 6 -b 320 -z 1 -L -I 3000 -y -u -t -R -T
```

Metadata file and directory removal test:

```
# mpirun -np 32 --nolocal --hostfile ./hosts /nfs/share/mdtest -d
/nfs/share/filedir -i 6 -b 320 -z 1 -L -I 3000 -y -u -t -r
```

B.3. Checkstream

The `checkstream` utility is available at http://sourceforge.net/projects/checkstream/. Version 1.0 was installed and compiled on the NFS servers and used for these tests.

First, a large file was created using the `genstream` utility. This file was copied to and from the NFS share by a client using `dd` to mimic write and read operations. Failures were simulated during the file copy process and the NFS service was failed over from one server to another. The resultant output files were checked using the `checkstream` utility to test for data correctness and ensure that there was no data corruption.

Given below is sample output of a successful test with no data corruption.

```
checkstream[genstream.file.100G]: ------------------------------------------------
checkstream[genstream.file.100G]: valid data for 107374182400 bytes at offset 0
checkstream[genstream.file.100G]: ------------------------------------------------
checkstream[genstream.file.100G]: end of file summary
checkstream[genstream.file.100G]: [valid data] 1 valid extents in 261.205032
seconds (0.00382841 err/sec)
checkstream[genstream.file.100G]: [valid data] 107374182400/107374182400 bytes (100
GiB/100 GiB)
checkstream[genstream.file.100G]: read 26214400 blocks 107374182400 bytes in
261.205032 seconds (401438 KiB/sec), no errors
```

For comparison, here is an example of a failing test with data corruption in the copied file. For example, if the file system is exported via the NFS async operation and there is an HA service failover during a write operation, data corruption is likely to occur.

```
checkstream[compute-00-10]: ------------------------------------------------
checkstream[compute-00-10]: valid data for 51087769600 bytes at offset 45548994560
checkstream[compute-00-10]:
checkstream[compute-00-10]: ------------------------------------------------
checkstream[compute-00-10]: end of file summary
checkstream[compute-00-10]: [valid data] 1488 valid extents in 273.860652 seconds
(5.43342 err/sec)
checkstream[compute-00-10]: [valid data] 93898678272/96636764160 bytes (87 GiB/90
GiB)
checkstream[compute-00-10]: [zero data] 1487 errors in 273.860652 seconds (5.42977
err/sec)
checkstream[compute-00-10]: [zero data] 2738085888/96636764160 bytes (2 GiB/90 GiB)
checkstream[compute-00-10]: read 23592960 blocks 96636764160 bytes in 273.860652
seconds (344598 KiB/sec)
checkstream[compute-00-10]: ------------------------------------------------
checkstream[compute-00-10]: encountered 1487 errors, failing
```

B.4. dd

`dd` is a Linux utility provided by the coreutils rpm distributed with RHEL 6.1. It is used to copy a file. The file system was mounted at `/mnt/xfs` on the client.

To write data to the storage, the following command line was used.

```
# dd if=/dev/zero of=/mnt/xfs/file bs=1M count=90000
```

To read data from the storage, the following command line was used.

```
# dd if=/mnt/xfs /file of=/dev/null bs=1M
```